Explorer++ Documentation

Release 1.3.5

Allen Titley

Nov 14, 2019

Contents

CHAPTER 1

About Explorer++

Explorer++ is a free multi-tab file manager for the Windows operating system, released as open source under the GNU GPLv3 license. Windows Explorer provides basic functionality - the ability to navigate throughout your computer, and perform basic tasks - but **Explorer++** goes further, introducing several enhancements and improvements for a much richer file browsing experience. Aside from basic navigation, **Explorer++** has:

- a two paned interface, like Windows Explorer, but with multiple tabs in the files pane. You can have many locations open at once and quickly change from tab to tab.

- a *Drives toolbar*, showing all available volumes - disk drives - (including network and USB drives). Jump to a drive with a single mouse click.

- a *Bookmarks menu* (similar to Windows Explorer favorites, but with more functionality), and a *Bookmarks toolbar*

- an *Application toolbar* where you can create shortcuts to your favorite applications

- *Show Command Prompt* built into **Explorer++**

- *attribute control* (set, clear) for multiple files

- various *methods* of file display, including multiple *groupings*

- quickly *split* a large file into pieces, for email or other purposes, and quickly *merge* pieces back into the original file

- a *destroy files* function which deletes files, then wipes their former disk space clean. This provides secure erasing of files.

- a built-in file *search* function which allows searching by name or attributes

- customized file display *colors*, by name or attributes

- ability to run in *portable mode*, storing all configuration in an XML file

- highly *configurable*, for user preference

- full drag-and-drop support with other applications, including Windows Explorer

- much more. . . !

Explorer++ requires Windows XP, Vista, 7, 8 or 10 (32-bit or 64-bit systems) to run; there is no installation required.

Explorer++ uses English as its native language, but can be used with other *languages* through the use of translation DLLs, available on the website. Translations are user contributed (thanks for the work!), and as a result, some may be incomplete.

Menus

2.1 File

2.1.1 New Tab

`Ctrl + T`

This function creates a new tab in the files pane. The tab is always created:

- as the last tab (far right) - default, or

- next to (right of) the current tab

depending on the setting in *Options.. (Tabs)*. The contents default to the current setting for same - see *Default new tab folder* option. The new tab automatically receives the focus (ie. becomes the active tab).


Tip: This function - opening a new tab - is also available by:

- using the middle mouse button while hovering over

  - a folder in either the *treeview* or files pane

  - a drive on the *Drives toolbar*

  - an item on the *Bookmarks toolbar*

- selecting **New Tab** (default contents) or **Duplicate Tab** (focus not transferred to new tab) from the tab's *context* (ie. right-click) menu

- selecting **Open in New Tab** from a folder's context (ie. right-click) menu in the files pane, or from a button's context menu on the *Bookmarks toolbar*

- pressing **Ctrl+Enter** while a folder in the files pane is selected

- using **Ctrl+***doubleclick* while hovering over a folder in the files pane

- double-clicking empty space in *Tab bar* (default contents)

- clicking the **Create a new tab** *button* (if visible) on the *Main toolbar*
- selecting **New Tab** from a Windows 7 *jump list* (available as a context, ie. right click) menu from the Task Bar

The above methods, unless stated otherwise, open the tab with the contents of the folder used, not the default contents or *Default new tab folder* (see above).

Windows 7 *Jump List*
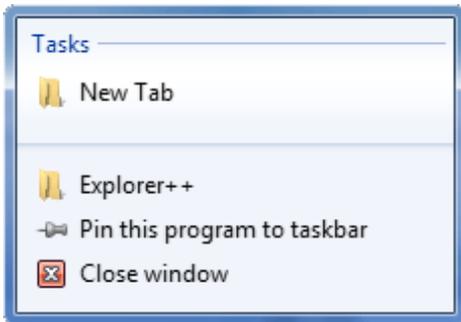
2.1.2 Close Tab

```
Ctrl + W
```

```
Ctrl + F4
```

This menu item closes the current tab in the file pane. Currently, there is no way to reopen a closed tab (ie. undo the closing operation), except using normal methods for a *new tab*.

Note: Starting with version 1.2, closing the last tab can close **Explorer++**, as an *option*.

2.1.3 Clone Window

This operation opens a new **Explorer++** window (instance), complete with all settings and panes copied from the original window. Clone Window only clones the active tab; other (non-active) tabs are ignored.

Clone Window obeys the setting of the *Allow multiple instances* option; if multiple instances are not allowed, then only a new tab is created and duplicated.

Note: Some settings (eg. column widths in the files pane) may not be duplicated.

2.1.4 Save Directory Listing. . .

This menu item performs a simple directory (folder) listing and saves it as a text file. A *Save As* Windows dialog allows the user to save the file in any location, under any name.

The listing only contains file/folder names, sorted in alphabetical order; folders are identified with "(Folder)" appended. Note that the file is saved in Unicode format (codepage = UTF-16 LE), with a BOM (byte order mark = FFFE hex) header. Windows Notepad can read this file format and can save the file is ANSI format, if desired.

Example

```
Directory
---------
H:\Projects\Explorer++\html\mnu_file

Date
----
2011-10-30, 7:39:58 AM

Statistics
----------
Number of folders: 0
Number of files: 15
Total size (not including subfolders): 42.7 KB

Folders
-------

Files
-----
clone_win.htm
close_tab.htm
copy_col_text.htm
copy_file_paths.htm
copy_folder_path.htm
copy_Ufile_paths.htm
delete.htm
delete_perm.htm
exit.htm
new_tab.htm
properties.htm
rename.htm
save_dir_list.htm
set_file_attr.htm
show_cmd_prompt.htm
```

Tip: More detailed lists can be obtained by opening a *Command Prompt* window and using the Dir command.

Example: `dir /ogn > "Directory Listing.txt"`

Type: dir /? for help on the **Dir** command.

2.1.5 Show Command Prompt

This menu item opens a command prompt (ie. *DOS box*) in the current directory. Executing command line commands can perform some tasks quickly and easily. Note that this opens a 32 bit (ie. **cmd.exe**) window rather than a 16 bit (ie. **command.com**) window.
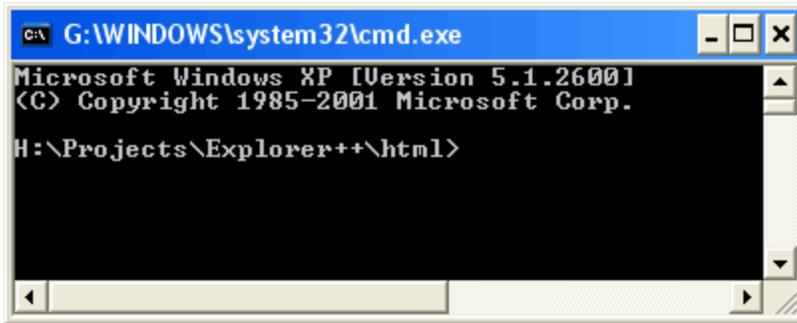
Note: For more information, search the Windows internal *Help and Support* for "command shell" and "command line reference".

2.1.6 Copy Folder Path

`Ctrl` + `Shift` + `P`

This function places the full (ie. *fully qualified*) path to the current folder - whose contents are shown in the files pane - on the clipboard, for pasting into another application. This may be useful for pasting into text or word processing documents. This is equivalent to

- selecting the path shown in the *address bar*, then

- copying that to the clipboard, using the context menu, or **Ctrl+C**.

The folder path is copied to the clipboard as Unicode text (ie. *plain* text) - the clipboard contents cannot be used to paste the **actual** folder or its contents. Because this is a Unicode copy, foreign characters, such as the *Beta* in the example, are represented correctly. If Asian characters are included, your Windows system must have support installed for "East Asian Languages" (*Control Panel/Regional and Language Options*). In order to save true Unicode information, pasting must be done into a text editor which is Unicode compatible, like Windows Notepad.

Example

```
H:\Projects\Explorer++\html\mnu_file\New ß Folder
```

Note: Using this function on a *virtual* folder (eg. My Computer, not really a true directory) copies the path as a special *registry path* or GUID (globally unique identifier) - not the path as shown in the *Address Bar*.

2.1.7 Copy File Paths

This function places the full (ie. *fully qualified*) paths to all currently selected files - as shown in the files pane - on the clipboard, for pasting into another application. If multiple file paths are selected, the paths are delimited (ie. separated) by carriage-return/linefeed characters; pasting multiple files to a text document pastes them as separate line.

File paths are copied as Unicode text - see *here* for more comments.

Folder paths, ie. selected folders displayed in the files pane, are copied the same as files.

Example
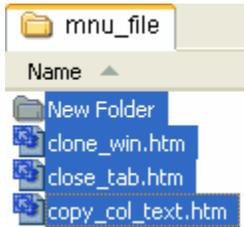

```
H:\Projects\Explorer++\html\mnu_file\New Folder
H:\Projects\Explorer++\html\mnu_file\clone_win.htm
H:\Projects\Explorer++\html\mnu_file\close_tab.htm
H:\Projects\Explorer++\html\mnu_file\copy_col_text.htm
```

Note: Network resources - files, folders and printers, etc. on a network - are copied using *UNC* rules, rather than fully qualified paths on the local computer.

2.1.8 Copy Universal File Paths

This menu item behaves nearly identically to *Copy File Paths*, except that the paths to Network resources (ie. files, printers, etc.) are always copied obeying Windows UNC (Universal Naming Convention) rules.

UNC syntax for Windows systems has the generic form:

```
\\ComputerName\SharedFolder\Resource, or
\\Server\Volume\File
```

A sample listing (1 file) might look like

```
\\Netbook\SharedDocs\My Pictures\Josh & Penny - house\IMG_0595.JPG
```

2.1.9 Copy Column Text

The files pane displays file and folder information in columns, such as:

The Copy Column Text function copies to the clipboard (as plain, Unicode text)

- column headings and

- the text shown in each column, whether visible or not

for files that are currently selected. Copied text may be pasted into another application, such as a word processor or text editor, using **Paste** (**Ctrl+V**) from within that application. Text columns are delimited by tab characters (09h), but columns may not *line up* in your word processor or text editor, depending on the length of text in each column.

Example

In the image above, using this function copies the following text to the clipboard:

```
Name Type Size Date Modified
about.png PNG Image 14.8 KB 8/4/2010, 9:44:56 PM
about-sample.gif GIF Image 247 bytes 8/6/2010, 10:31:02 AM
```

2.1.10 Set File Attributes...
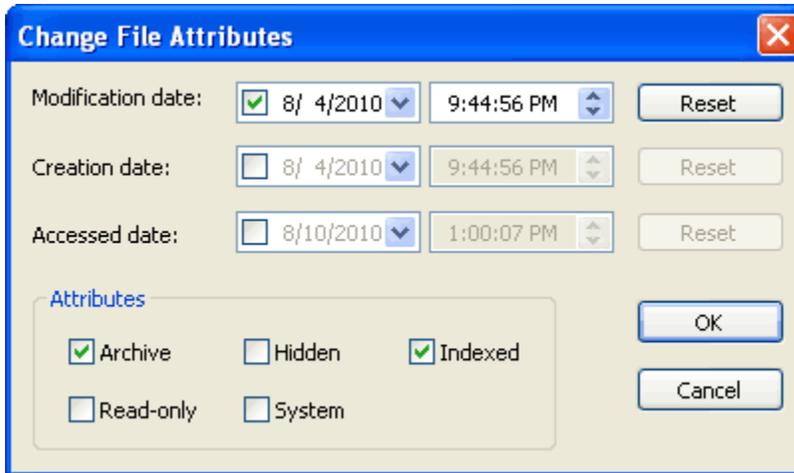
This dialog allows the user to change the file attributes, including system time stamps, of selected files and folders.

Change date manually

You may select each of the three fields (month/day/year, shown here), then type a new value. For example, selecting the day field and typing 23 changes the day to the 23rd of the month shown.

Note: The individual fields may not be the same as shown here - they are determined by the setting for Short date format, found in the *Control Panel* / Regional and Language Options applet.

Calendar tool
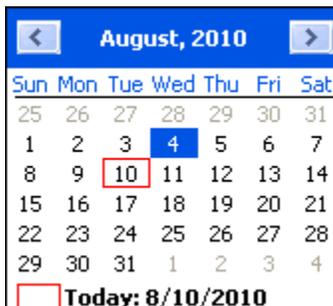
The calendar tool allows you to quickly select a date. You may

- click any **day** of the month to select the day

- cycle through **months** by clicking the left/right buttons

- click the **month** (eg. August) for a *drop-down* list of months and select one

- click the **year** (eg. 2010) for a *spinner* control to adjust the year

- click the **Today:** line at the bottom to select *Today* (ie. today's date)

Change time manually

As with the manual date change, you should select a field, then type a new value. Type "A" or "P" to change to AM or PM.

Attributes

The checkboxes may be checked (set attribute) or cleared (clear attribute) to affect the following attributes:

Archive

The archive attribute is used to determine whether a file or folder has been changed. Whenever a file (not folder) is created or saved, the archive attribute is set. Most backup programs may be configured to clear the archive attribute as files are backed up.

Read-only

This attribute marks a file or folder to prevent overwriting or deletion. Obeying the status of the **Read-only** attribute is the duty of individual programs using those files; some may ignore any read-only status. In general, the Windows system will warn you before doing deletion or overwriting of **Read-only** files.

Hidden

Assigning the hidden attribute flags a file or folder to prevent it from *being shown* during normal file listings, providing **Explorer++** (or Windows Explorer) is not set to view hidden files. Command line operations (eg. `Dir`) also obey the setting of the hidden attribute.

System

Windows tries to protect system files (necessary for system operation) in a number of ways, one of which is to assign the system attribute. When this attribute is applied, some operations will respect it and alter their function, ie. refuse deletion. In most cases, this behavior can be over-ridden. Windows can also set the **Hidden** attribute on a system file, making it *superhidden*.

Indexed

Starting with Windows 2000, the indexing service creates indices of files for speed up of searches. With this attribute set, the file is marked for inclusion in the index.

Note: The **Date Accessed** (or **Accessed date** in the dialog) attribute may be considered to be unreliable as:

- it is not present on FAT16 file systems,

- has no time component (ie. resolution of 24 hours) on FAT32 file systems,

- may not be transferred correctly from system to system, and

- may not be enabled on systems running Windows XP. See *here* for a tip on enabling this feature.

The **Indexed** attribute is not supported by FAT16 and FAT32 file systems.

2.1.11 Delete

`Delete`

This function deletes - with a confirmation dialog - selected files and folders. Deleted items are sent to the Windows Recycle Bin and may be recovered from there, if necessary.

Read-only files or folders are deleted as usual, but are identified as read-only by the dialog; if a read-only file is among the selected files (not read-only) it will require an additional confirmation.

System files receive a similar warning, but may still be deleted (probably not a good idea!).

Note: Files deleted from removable drives (eg. *flash* or *jump* drives are not sent to the Recycle Bin but are *deleted permanently* by Windows. Not all USB drives are considered removable drives; some larger drives install as local drives. Viewing the *properties* for a drive in the Folders pane will identify it as Local (fixed) or Removable.

2.1.12 Delete Permanently

`Shift + Delete`

This menu item *deletes* selected files and folders, but does **not** send them to the Windows Recycle Bin. This is not a *secure* delete; see *Destroy File(s)...* for secure deletion of files.


Note: Files and folders deleted by Windows are not really removed (immediately) from the file system. Instead, the *File Allocation Table* (FAT16 and FAT32 file systems) or *Master File Table* (NTFS) is marked indicating that the file has been deleted, and its entry is cleared. The disk space occupied by the file is declared free for use, but the actual file data is not immediately removed. It can, however, be over-written; once that occurs, the file is usually unrecoverable, except possibly in part, by recovery utilities such as Recuva, a freeware product by Piriform.

2.1.13 Rename

`F2`

This routine is really two routines in one, depending on whether a single file (or folder) is selected or *multiple* items are selected.


Single item
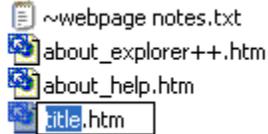
Explorer++ allows you to edit the name in an *in-place* edit box, keeping the extension the same. The extension may be edited, but must be deleted then re-typed or selected then over-written.

Tip: Pressing F2 during a single item rename cycles between highlighting

- the file name

- the file extension the full name, including the extension

Multiple items - Mass Rename
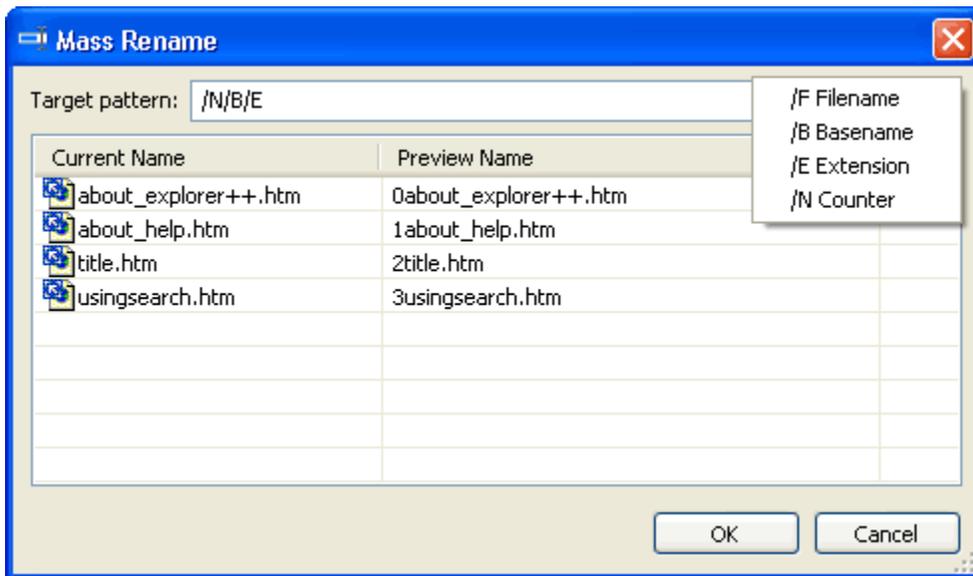
The **Mass Rename** dialog allows you to build a new name for all files selected using alphanumeric characters, combined with special character combinations or *words*. The special *words* are visible in a *reminder* box by clicking the arrow button to the right of the **Target pattern:** edit box. Note that special *words* are case-sensitive and any characters that do not form a special *word* are placed in the new filename *as-is*.

/F	resolves to the entire filename, as originally named. Same as **/B/E**
/B	resolves to the filename minus extension, example: for "about_help.htm", **/B** resolves to "about_help"
/E	resolves to the file extension, including " **.** ", example: for "title.htm", **/E** resolves to ".htm"
/N	resolves to an integer, beginning with 0 for the first file

Typing the new file name pattern into the edit box shows the new files previewed to the right of the existing names. Clicking OK completes the rename operation.

Note: *Undo* works on the entire list as a set; all files renamed by a Mass Rename as restored to their original names.

2.1.14 Properties

`Alt` + `Enter`
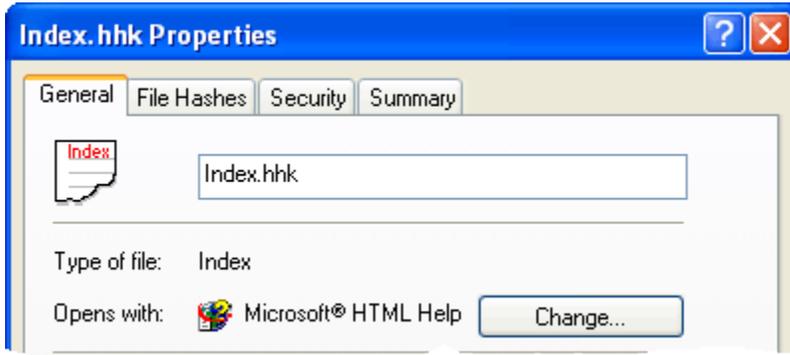
This function opens the Windows property sheets for the file or group of selected files. Property sheets may vary, depending on your configuration, and whether additional utilities are installed, but in general should show information such as:

- **Type of file:** as recognized by Windows, eg. *Text document* (.txt), *PNG image* (.png), etc.

- **Opens with:** an application associated with the file, if any

- **Location:** folder containing the file

- **Size:** and **Size on disk:** actual file size (of contents, if folder is selected) and disk space used

- **Contains:** number of files and folders contained in a folder (if a single folder is selected only)

- *File attributes* currently applied

- **Security** settings and permissions (NTFS only - may not be shown, depending on system settings)

- File **Summary:** comments, if any assigned to the file by the user or an application (NTFS only)

- **Customize:** allows you to change certain properties (eg. image, icon) of a folder or group of folders

Utilities such as Hashtab (which displays *hashes* for the file, such as CRC32 and MD5) may introduce additional property sheets.

If multiple files are selected, many of the same properties are shown, including **Size:** and **Size on disk:**, which now refer to the collective size of all the files selected.

The **Properties** function is also available as part of the context (ie. right-click) menu in the folder or files panes.

Note: The Properties dialog can also be activated by **Alt**+*doubleclick* while hovering over a file or folder.

2.1.15 Exit

This menu item exits **Explorer++**. Prior to closing, **Explorer++**

- saves settings to an XML file (config.xml) in the same folder as *Explorer++.exe*, or

---

- saves settings to the Windows registry,

depending on the setting for *portable* mode.

The window close button and closing the last remaining tab in the files pane (subject to an *option*), perform the same operations as **Exit**.

2.2 Edit

2.2.1 Undo

`Ctrl + Z`

This function allows you to undo a **single** file or folder *renaming* operation, restoring the original name. When renaming multiple items using the *Mass Rename* dialog, all items can be undone as a single operation.

> **Caution:** **Explorer++**'s **Undo** cannot undo a file cut, move or delete.

Note: Renaming a folder which has been *bookmarked* does **not** automatically rename the bookmark.

2.2.2 Cut

`Ctrl + X`

The **Cut** operation removes selected files and folders and places them on the clipboard, for *pasting* in a different location. **Cut** may be used from the files or folders pane.

Items to be cut are still in their original location - they are only *marked* for cut - until pasted, at which time the originals are removed. Items cut are **not** placed in the Recycle Bin; they are *deleted permanently* from their original location after pasting.

2.2.3 Copy

`Ctrl + C`

Copy places selected items (files and folders) on the clipboard for subsequent *pasting*. The clipboard's contents may be pasted into the source folder; copies of all items are created.

Copy does not delete the original files or folders.

Note: Copying (and *Pasting*) files or folders from other locations (ie. not the same folder) detects *collisions* (duplicate names) and presents the *usual* dialog allowing replacement, if desired. Files or folders copied from the **same** folder are renamed - Copy of. . . , Copy (2) of. . . , etc.

2.2.4 Paste

`Ctrl + V`

This menu item takes data from the clipboard and pastes it into the files pane, or the folders pane if it has the focus. If files or folders were *cut* in order to place them on the clipboard, then they are deleted at this time. **Explorer++** handles various types of data, and the pasting method, differently, according to the following table:

Clip-board data	Paste method	Result
files/folders	paste into blank area of files pane	files/folders pasted normally to location of the files pane
files/folders	paste into folders pane (using menu item, or toolbar button)	files/folders pasted into the folder having the focus in the folders pane. New files/folders will, of course, appear in the files pane (contents of focused folder).
text	paste into dialog, or rename file operation	text inserted, or overwrites text selection
text	paste into blank area of files pane	text pasted as new text file, automatic naming: eg. Clipboard Text (10-20-2010, 6.44.01 PM).txt
text	paste into folders pane (using menu item, or toolbar button)	text pasted as new text file (as above) into the folder having the focus in the folders pane. New file will, of course, appear in the files pane (contents of focused folder).
bitmap (im-age)	image data, copied from a paint program, for example is pasted into blank area of files pane	new image created, automatic naming: eg. Clipboard Image (10-20-2010, 6.53.48 PM).bmp New image always created as Windows bitmap, regardless of original image type.

See *Copy* for a discussion of behavior when file names are duplicated.

2.2.5 Paste Shortcut

`Ctrl + Shift + V`

This menu item pastes Windows *shortcuts* to the files and folders currently on the clipboard, in the current folder as shown in the files pane. The files and folders would have been placed on the clipboard by *Copy* or *Cut*. If **Cut** was used, then pasting shortcuts does **not** delete the original file, as with *Paste*.

About Shortcuts

Shortcuts are small files (typically < 1000 bytes) - always with the extension **.LNK** - which contain, among other things, the path to a file or folder. A shortcut file behaves, for the most part, the same as the file it references. That is,

- double-clicking a shortcut (or .LNK file) opens the target file

- using a context (or right-click) menu to Edit a shortcut, opens the target file for editing

- etc.

Creating shortcuts using this function always creates a .LNK file with the *fully qualified* path (eg. C:\folder\file.ext) used, rather than a *relative* path (eg. ....\folder2\file2.ext), and may, therefore, be moved anywhere on the local computer and will still function correctly.

Since shortcuts are true files and only link to the target through a (text) path, deletion or modification of a shortcut has no effect on the target file.

A good use for shortcuts is to place in one location (ie. one folder) a collection of shortcuts to files which are related in content or purpose, but which are scattered over the whole computer.

Shortcuts may be edited by using the *Properties* function. Typically, the user may change

- the shortcut's name. When created, it has the same name as the target file, but this is not necessary.

- the application used to open the target file

- the shortcut's (not the target file's) *attributes*

- the target file, working folder (if any) and hotkey (if any)

- the shortcut's icon (default is the icon of the target file)

- security settings (ownership, permissions, etc.), if implemented (NTFS drives only)

Shortcuts may be created (by other means, such as <right-click> *Create Shortcut*) to Windows special folders, such as *Control Panel*, which are not true physical folders. The target for these types of shortcuts is usually not editable. Also, installation of some applications may create shortcuts to those applications which are not editable.

2.2.6 Paste Hard Link

This menu item pastes Windows *hard links* to the files currently on the clipboard, into the current folder as shown in the files pane. The files would have been placed on the clipboard by *Copy* or *Cut*. If **Cut** was used, then pasting shortcuts does **not** delete the original file, as with *Paste*. Note that folders **cannot** have hard links.

About Hard Links

Hard links are a feature of the NTFS file system (ie. not available on FAT16 or FAT32 drives) which allows creation of *virtual files* which point to a true file. Multiple *virtual* files may point to the same file, with each behaving as though it were the true file. The hard link *points* to the MFT (Master File Table) entry for the true file and propagates changes to the entry (except for the file name) to all other hard links *pointing* to the same file.

The following diagram may explain this concept better, but may not be completely accurate in its representation of the MFT entry or file structure.
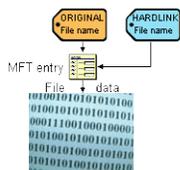
In the diagram, the original name points to (and accesses) an MFT entry which contains all file attributes (timestamps, read-only flag, etc.) and in turn points to the actual file data. When the hard link is created, it points to the **same** MFT entry, but could have a different name. The **Explorer++** function - *Paste Hard Link* - always creates the hard link with the original file name; obviously, then it cannot be created in the original folder or else a naming conflict will exist. Since the original name and hard link name use the same MFT entry, etc., changes to either file result in changes to the true file data and attributes; the newly created hard link *file* is essentially a *true file*! To remove a hard link, deleting either file name has the same effect; the remaining (single) link becomes the *true file*.

In Windows, the NTFS file system counts the original name as hard link #1 and the additional hard link as #2; all *normal* files, then would have a hard link count of 1. The file system tracks the hard link count for each file at all times; when the count becomes 0, then the file data area is considered free space and may be reclaimed. It is interesting to note, therefore, that when a file is deleted (to the Recycle Bin), it would still have a count of 1 since only the name (and MFT entry?) have been removed. It is not until the Recycle Bin is emptied that the file is completely deleted, assuming its hard link count was 1, ie. no additional hard links to the file exist.

Hard links may **only** be created for files on the same volume (ie. the same drive letter); attempting to paste a hard link to a file on a different volume results in a true copy of the file.

Explorer++ can view the hard link count by enabling *Hard links* as a *column heading* in the files pane.

If you use hard links frequently, you might consider a useful freeware tool - Link Shell Extension - which can manage hard links. When installed, it (among other things) creates an additional *property sheet* which allow enumeration (ie. location) of any additional hard links to a file, something that Windows can't do!

Note: Hard links may also be created by the Windows command line tool `Fsutil` (Windows XP) or `Mklink` (Windows Vista or later).

2.2.7 Copy To Folder

`Ctrl` + `Shift` + `C`

This menu item copies selected files and folders to a different location. After selecting the files/folders and selecting this item, the user is presented with a standard *Browse for folder* dialog (see *here* for a picture of this Windows dialog), allowing the destination location to be chosen.

Using this function is the same as a *Copy*/*Paste* operation, but without changing tabs or folders on a tab.

2.2.8 Move To Folder

`Ctrl` + `Shift` + `M`

This menu item moves selected files and folders to a different location. After selecting the files/folders and selecting this item, the user is presented with a standard *Browse for folder* dialog (see *here* for a picture of this Windows dialog), allowing the destination location to be chosen.

Using this function is the same as a *Cut*/*Paste* operation, but without changing tabs or folders on a tab.

2.2.9 Resolve Link

This function *resolves* (ie. locates) the target file (or folder) of a **single** selected *shortcut*. A new tab is opened and the target file is selected. Locating the file/folder in this way is usually considerably faster than navigating to the target manually.

If the target cannot be located, for example, if it has been deleted, a Windows error message is displayed which allows you to locate the target file.
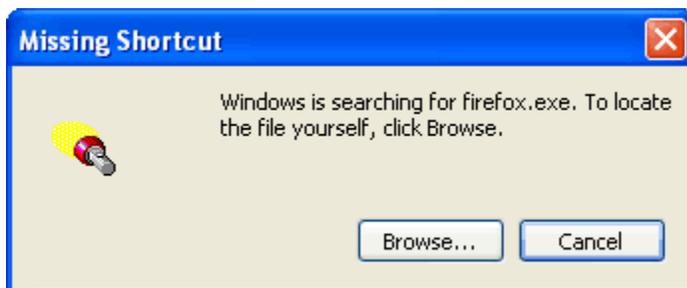
2.3 Selection

2.3.1 Select All

`Ctrl` + `A`

This function selects **all** files (and folders) in the files pane as shown in the current tab.

Note: Individual files may be deselected, after using **Select All**, by using **Ctrl+**_click_on_file_. In *Details* view, the file must be clicked in the first column, unless the *Full row selection in details view* option is enabled.
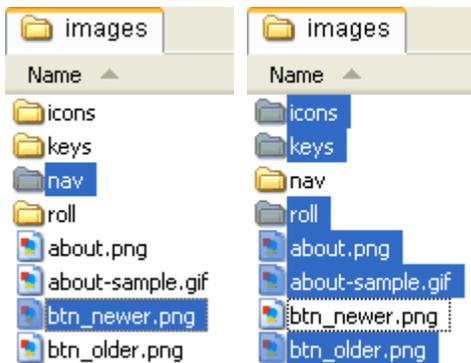

2.3.2 Invert Selection

`Ctrl + I`

Using this function affects the files or folders in the files pane, causing

- selected files to be deselected, and
- deselected to be selected.

In other words, it inverts that state of selection for all contents of the current tab.

2.3.3 Select All Of Same Type

This function selects all files in the folder of the same type as the **one** currently selected. For example, if the files pane (ie. currently folder) shows multiple types of images, ie. *gif*, *png*, *jpg*, *bmp*, etc., it is possible to select only jpg images by

- select a single jpg image, by clicking on it
- use this function - **Select All Of Same Type**.

The end result is that all jpg files will be selected.

Tip: **Select All of Same Type** uses the file *type*, as determined by the Windows registry - not the file extension! This means, for example, that *image1.jpg* and *image2.jpeg* will **both** be selected, since they are both of type **JPEG Image**. Windows file types may be viewed by *enabling* the *Type* column in the files pane. You may select by file name using the *Wildcard Select. . .* routine.

2.3.4 Select None

`Ctrl + Shift + N`

This menu item deselects **all** files (and folders) in the files pane, ie. current tab.

All files may also be deselected by clicking in a blank area in the files pane, in the first column (usually *Names*) or clicking in any other column.

Note: Individual files may be selected, after using **Select None**, by using **Ctrl**+*click_on_file*. In *Details* view, the file must be clicked in the first column, unless the *Full row selection in details view* option is enabled.

2.3.5 Wildcard Select. . .

`Ctrl + Shift + S`

This menu item allows the user to select files in the files pane (ie. current folder) using a wildcard pattern. A dialog is opened which allows the user to enter a new wildcard pattern, or use the *drop-down* button and choose a previously used pattern. Clicking OK selects files and folders based on that pattern. Repeated use of **Wildcard Select. . .** is cumulative; additional files may be selected, adding to the current selection.

Unlike *Select All Of Same Type*, the selection process is based on file name and extension.

About wildcard patterns

A wildcard pattern is a file specification which contains symbols representing unknown characters, as well as *ordinary* alphanumeric characters.

The two wildcards available here are:

- *** (asterisk)**

  This character substitutes for zero or more characters. Some examples are:

  *file.txt matches "somefile.txt", "my_file.txt", but not "my_files.txt" (intervening "s" prevents match).

  btn*.* matches "btn_23.png", "btnumbers.txt" and "btn-img.jpg", but not "mybtn.jpg" (leading "my" prevents match) or "btn12" ("." specified must exist)

- **? (question mark)**

  This character substitutes for any single character, which must exist. Some examples are:

  final?.doc matches "final1.doc", but not "final_chapter.doc" (? matches only 1 character, "_chapter" is not 1 character)

  file.tx? matches "file.txt", but not "file.tx" (last wildcard character must exist)

2.3.6 Wildcard Deselect. . .

`Ctrl + Shift + D`

This menu item allows the user to deselect files (already selected) in the files pane (ie. current folder) using a wildcard pattern. A dialog is opened

---

which allows the user to enter a new wildcard pattern, or use the *drop-down* button and choose a previously used pattern. Clicking OK deselects files and folders based on that pattern. Repeated use of **Wildcard Deselect. . .** is cumulative; additional files may be deselected, reducing the current selection.

Wildcard Deselect. . . deselects files (and folders) based on file name and extension.

See *Wildcard Select. . .* for examples of wildcard patterns.

2.4 View

2.4.1 Status Bar
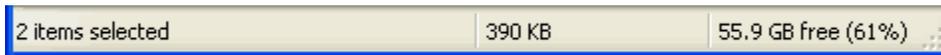
The status bar presents some quick information and may be toggled on/off from the **View** menu.

Items/text

This area shows either

- the number of *items* (files and/or folders) and displays one of the following:
    - if no selection exists - total **item count** (eg. *24 items*), a count of items in the folders pane (current folder)
    - if a selection exists - total **selection count**, the number of items selected, or
- the *helper text* of a menu item when the mouse is hovered over that item.

    Example: hovering over the File/*New Tab* menu item displays "Creates a new tab".

Size

This area shows the actual item (files and/or folders) size and displays one of the following:

- if no selection exists - total **file size**, the combined size of all files in the folder. Note that folder items are not included in this size, even if the *Show folder sizes* option is enabled.
- if a selection exists - total **selected file size**, the combined size of all selected files (not folders)
- if the current tab is a *virtual* folder (ie. special Windows folder) - the words **Virtual Folder**. Selecting items has no effect.

If a real size is shown (not a virtual folder), then the size is affected by the *Show all file sizes in:. . .* setting.

Free space

This area shows free disk space for the volume which houses the currently active folder (files pane). Free space is shown in bytes (rounded to MB, GB, etc., regardless of any options settings) and in percentage of the entire volume.

2.4.2 Folders

This menu item toggles the visibility of the Folders pane. When the **Folders pane** is hidden, its former window area is used by the Files pane, allowing more columns (or wider columns) to be displayed.

Toggling the visibility of the **Folders pane** may also be done through a button on the *Main toolbar*.

2.4.3 Display Window

This menu item toggles the visibility of the Display Window. The Display Windows presents some additional information about selected files.

2.4.4 Toolbars
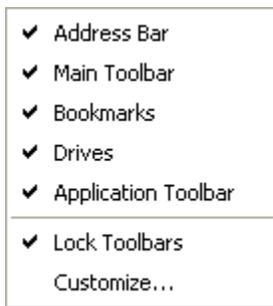
About toolbars

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.

2.4.5 View Modes

The Files pane may present files and folders using a variety of display methods. Each has its own merit, depending on the content being displayed, and may display different amounts of file information.

Views may be also accessible from a button on the *Main toolbar*.

Note: In most views, the name may appear to be the primary display item. Actually, it the leftmost information column (as shown in *details* view) which is displayed! In *icons* view, for example, the file size could be displayed instead. See *Select Columns. . .* for more information.
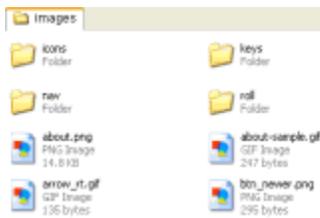

Thumbnails

This view displays thumbnail images for each file. Using **Thumbnails** is primarily useful when displaying images; other files may simply display a *generic* image assigned by Windows.

Explorer++ presents a view of each folder, as well, showing a few thumbnails of images (if any) contained therein.

Currently, the thumbnail size is fixed (internal to **Explorer++**) at 120x120 pixels. **Explorer++** does **not** cache thumbnails as Windows Explorer does (optionally creates "Thumbs.db" file).
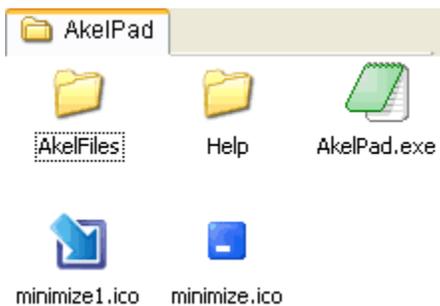
Tiles

Tiles view displays the files and folders, using the Windows *generic* images for file types. In addition, the file type and size is displayed.

Files and folders are displayed in a manner similar to **Thumbnails** - in as many columns as the width of the Files pane will allow.

Icons

Icons view uses embedded (resource) icons from files and, as with **Thumbnails** and **Tiles**, displays the files and folders in columns, as space allows.

Icons view is perhaps better suited to displaying executable files, and icon files (of course). Icons are displayed in 32x32 bit size.

Note: Windows Vista / 7 presents 4 different sized icons for *View* choices.
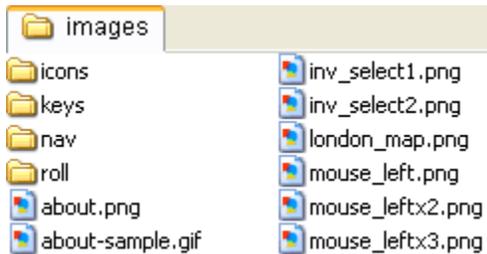

Small Icons

The **Small Icons** view is similar to **Icons** view, but the icons are displayed in 16x16 bit size allowing more files/folders to be displayed in the same space.
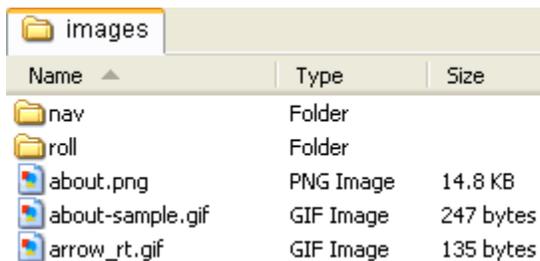

Bug

The **Small Icons** view currently has some acknowledged deficiencies in layout behavior.

List

List view displays files using *generic* icons and a single piece of information, usually the file name (see *above*). Items are displayed in columns to fit the available space.

Details

The **Details** view shows textual information and generic icons, in a single vertical list. It can display multiple columns of information, as *set* by the user. This display method can yield the most information, but usually takes up the most pane width. Columns not displayed due to lack of pane width may be scrolled into view as needed.

Items displayed in this view may be re-arranged using drag-and-drop. This has no effect on their actual order in the folder, but may aid the user in selections, editing, etc.

2.4.6 Auto Arrange

When displaying items using the *Thumbnails*, *Tiles*, *Icons* or *Small Icons* methods, items are normally displayed in *orderly* columns, laid out according to available width in the Files pane. Checking the **Auto Arrange** menu item enforces this requirement.

When **Auto Arrange** is unchecked, items may be dragged to positions outside the columns allowing the user to visually organize the pane to his liking. This is not a permanent re-ordering; the effect is merely visual.

If the view is changed, for example from **Thumbnails** to **Icons**, the items are initially organized in columns even if **Auto Arrange** is not enabled.

The following image demonstrates what can be done with **Auto Arrange** turned off (disabled).
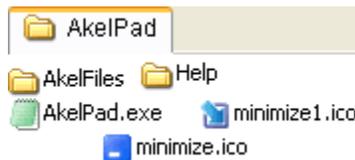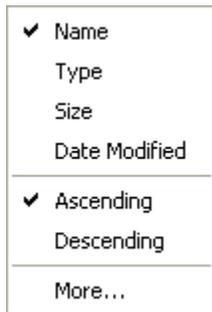

Fig. 1: Auto-arrange on

Fig. 2: Auto arrange off

2.4.7 Sort By

This menu selects a sort order for display of files and folders in the Files pane.

Selecting a **Sort By** order cancels a *Group By* order. The **Sort By** menu is built by **Explorer++** based on the columns currently enabled in *Details* view; your menu may differ from the one shown here. Selecting an item in this menu has the same effect as clicking on the header for that column in the Files pane.

Sort order

Current sorting target

The current sorting target must be an enabled column, which would be shown in *Details* view. This may be any column, but some are more useful that others. Sorting by **Name**, for example, places items in alphabetical order, while sorting using the **Owner** (as per Windows security and permissions settings) column may yield results which are essentially unusable, since likely all files have the same owner.

Following is a list of just a few common sorting targets available in the current version of **Explorer++**. See *Select Columns...* for a more complete list.

Name	This is the file name, including extension
Type	Windows registered file type (eg. JPEG image)
Size	File size (actual, in bytes)
Date Modified	Last date file was modified/changed
Attributes	eg. read-only, hidden, system
Extension	Just the file extension (eg. *txt*)

Note that the list of available sorting targets (ie. **Sort By** menu) may change depending the active folder. Some special folders, such as **My Computer**, for example, have a different list of sorting targets (or display columns) than others.

Sort direction

Items for display may be sorted in

- **ascending** order, that is, alphabetically - *a* before *b*, smaller before larger, etc., or

- **descending** order, that is, alphabetically - *b* before *a*, larger before smaller, etc.,

depending on your preference. For example, with **Date Modified** as the target, it may be more desirable to use **descending** order, thereby placing the most recent items at the top of the list. If you are trying to locate files with zero length, then using an **ascending** sort with **Size** as the target places zero length files at the top of the list.

Since the sorting *target* is always an enabled column (as would be shown in *Details* view), the column heading reflects both the target and direction, as shown in the vertical arrow.
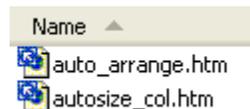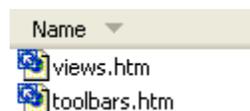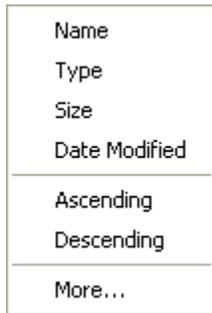

Fig. 3: Name/Ascending

Fig. 4: Name/Descending

More. . .

This menu item opens the *Select Columns. . .* dialog allowing you to enable additional columns, or disable existing ones. Once the columns are managed, the Sort By menu must be re-opened to set a different sorting *target*.

2.4.8 Group By

This menu selects a sorting order and grouping for display of files and folders in the Files pane.

Selecting a **Group By** order cancels a *Sort By* order. The **Group By** menu is built by **Explorer++** based on the columns currently being displayed in *Details* view; your menu may differ from the one shown here.

2.4.9 Show Hidden Files

```
Ctrl + H
```

Clicking on this item toggles display of hidden files - on or off. Hidden files are those which have the *hidden attribute* set; they are *usually* masked from display if marked as hidden. This setting can be changed on a tab-by-tab basis; new tabs are created with this setting enabled.

In the Files pane (only), hidden files and folders are displayed with their accompanying icons *ghosted* (ie. *washed out*).

The *Control Panel/Folder Options/View* setting - **Show (Do not show) hidden files and folders** - has no effect on display of hidden items (see *here*); **Explorer++** controls hidden visibility for itself.

Tip: System files and folders may be *superhidden* - that is, they have **both** the Hidden and System attribute set - and are not displayed, even if **Show Hidden Files** is checked. To allow *superhidden* files and folders to be affected by this function, you must uncheck the *"Hide protected operating system files"* setting in the *Files and Folders tab* of **Explorer++**'s options.

Display of *superhidden* files is contingent on a Windows system setting being enabled; you must uncheck the *"Hide protected operating system files"* setting in the **Folder Options** applet (*View* tab). The **Folder Options** applet is found in the **Control Panel**. Note that **Explorer++** can control visiblility of *ordinary* hidden files (Hidden attribute, but **not** System attribute) **without** enabling the *"Show hidden files and folders"* option in the **Control Panel**.


2.4.10 Refresh

```
F5
```

```
Ctrl + R
```

In the files pane, all files (depending on **Explorer++** setup) are displayed and sorted. Using **Refresh** updates the display; the files are re-sorted and re-displayed, subject to current settings (ie. *Sort By*, *Group By*, *Show Hidden Files*, etc.).

Refresh also restores the address of the currently active tab in the *Address bar*.

Note: The folders pane is not refreshed (ie. rebuilt) by **Refresh**. Collapsing (clicking the minus, eg.), then expanding (clicking the plus, eg.) for a *node* in the Folders pane will refresh that *node*.

2.4.11 Select Columns...
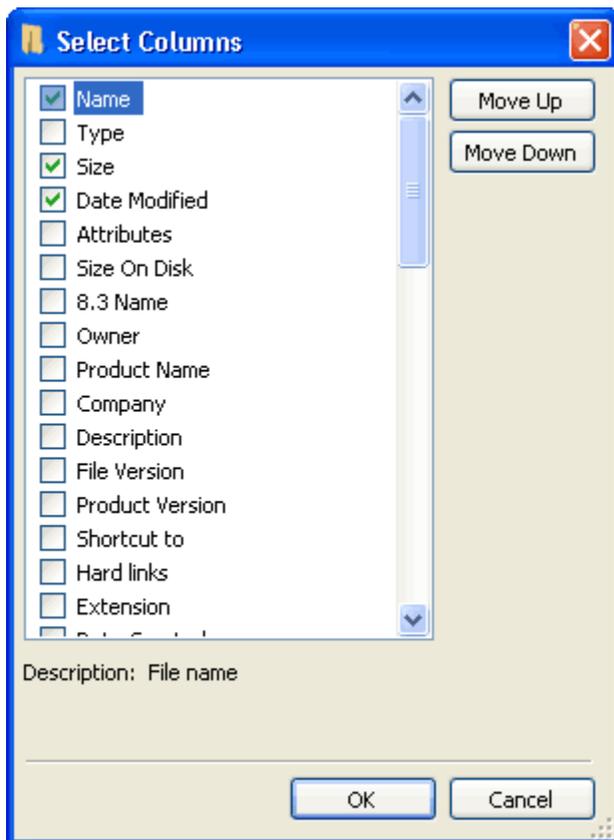
Explorer++ allows you to set data column visibility for folder listings on a **tab-by-tab** basis. Columns available also depends on the current tab's *folder type*. The settings here apply only to the current tab; future (new) tabs adopt the settings from the *Default Columns...* setting.

For more information about column settings, see *here*.

Columns available

This area shows the list of columns available for the *type* of folder currently being viewed; checking the checkbox beside the name enables the column in the Files pane. This list of available columns may vary with the type of folder currently being viewed (ie. the active folder for a tab). In addition, while a particular column may be checked, not every file in the folder may possess the type of data placed in that column. For example, image files may show data in

the **Width** and **Height** columns, but text files will not. In addition, the columns is different for virtual folders, etc, as outline in the lists following.

Column Order

Columns are displayed (left to right in the Files pane) in the order shown, from the top down. The order may be changed using the **Move Up** and **Move Down** buttons.

Folder type

Folder type/name	Comments
Control Panel	Windows *virtual* folder: a collection of *applets* for modifying your system, etc.
General	This is a normal (ie. real) folder on your system. This type of folder comprises 99% of what you will view and presents, therefore, the most common list of columns.
My Computer	Windows *virtual* folder: shows volumes (ie. drive letters) and some other virtual or system folders, such as *My Documents*, *Control Panel*, etc.
Network Connections	Windows *virtual* folder: This is actually the *Control Panel* applet which manages connections.
My Network Places	Windows *virtual* folder: This is actually the *Control Panel* applet which manages connections.
Printers and Faxes	Windows *virtual* folder: list of printers, faxes and printer drivers connected to your system.
Recycle Bin	Windows *virtual* folder: collection of erased files which are still recoverable.

Following is a list of all columns available for the different types of folders which can be viewed. Default columns (with a checkmark in the **Def** column) and default column order, assigned by the system (ie. order, left to right) ,are also shown. The user may change the default widths (always 150 pixels) through the *Save Column Layout as Default* menu item.

Incomplete

Columns marked as "**n/a**" (not available) may not be implemented as of this version of **Explorer++**, ie. they may be present but contain no data.

Control Panel

Def	Column	Description
✓ 1	Name	*applet* name
✓ 2	Comments	brief description of applet's purpose

General

Applies to *real* files or folders. Not all tags apply to all file types.

Def	Column	Description or Comment
✓ 1	Name	file or folder name
✓ 2	Type	Windows associated descriptive file type
✓ 3	Size	file size or folder contents size (if enabled)
✓ 4	Date Modified	date of last file/folder modification. Some applications may modify the file without changing this attri
	Attributes	common file *attributes*
	Size On Disk	amount of disk space used by file
	8.3 Name	file name in older (ie. DOS) 8 character name, 3 character extension format
	Owner	Windows security - owner of file or folder
	Product Name	executable - from Version *property sheet*
	Company	executable - as above
	Description	executable - as above
	File Version	executable - as above
	Product Version	executable - as above
	Shortcut to	target of shortcut (.LNK or .PIF) file
	Hard links	number of hard links associated with this file - see *here* for more
	Extension	file extension (minus *dot*)
	Date Created	date of file or folder creation
	Date Accessed	date of last access (read or write) of file or folder. On Windows XP this may only reflect the nearest ho
	Title	user supplied - title of file - entered in Summary property sheet
	Subject	user supplied - subject of document - entered in Summary property sheet
	Author	user supplied - author of document - entered in Summary property sheet
	Keywords	user supplied - associated key words - entered in Summary property sheet
	Comment	user supplied - comment - entered in Summary property sheet
	Camera Model	EXIF data, recorded by camera in an image file
	Date Taken	EXIF data, recorded by camera in an image file
	Width	width of image (bits)
	Height	height of image (bits)
		Note: The following media tags **may** be associated with audio, video or ? - tagging of media files allo
	Bit rate	media (audio/video) file - bit rate associated with data
	Copyright	media (audio/video) file - internal tag identifying file as copyrighted
	Duration	media (audio/video) file - play time
	Protected	media (audio/video) file - identifies file as protected media
	Rating	media (usually video) - internal tag for rating
	Album artist	media (audio) - internal tag (text)
	Album	media (audio) - internal tag (text)
	Beats-per-minute	media (audio) - internal tag
	Composer	media (audio) - internal tag (text)
	Conductor	media (audio) - internal tag (text)
	Director	media (video) - internal tag (text)
	Genre	media (audio) - internal tag (text)
	Language	media (audio) - internal tag (text)
	Broadcast date	media (video) - internal tag (text)
	Channel	media (video) - internal tag (text)
	Station name	media (video) - internal tag (text)
	Mood	media (audio) - internal tag (text)
	Parental rating	media (video) - internal tag (text)
	Parental rating reason	media (video) - internal tag (text)
	Period	media (video) - internal tag (text)
	Producer	media (video) - internal tag (text)

Co

Table 1 – continued from previous page

	Publisher	media (audio) - internal tag (text)
	Writer	media (video) - internal tag (text)
	Year	media (audio) - internal tag (text)

Tip: The *Date Accessed* timestamp may not be reliable in Windows XP. It is subject to a registry setting - *NtfsDisableLastAccessUpdate* - which does not exist, by default. To enable automatic updating of *Date Accessed,* open a command prompt window and execute the following command:

```
fsutil behavior set DisableLastAccess 0
```

Executing this command creates the required registry entry and enables automatic updating of *Date Accessed.* Updating can be disabled by setting the value to 1.

My Computer

Def	Column	Description
✓ 1	Name	drive name (and letter) or special folder name
✓ 2	Type	Local Disk, CD Drive, Removable Disk, Folder, etc.
✓ 3	Total Size	size available on drive
✓ 4	Free Space	unused drive space available (bytes, GB, etc.)
	Comments	provided by Windows - folder function, drive details, etc.
	File System	FAT (ie. FAT16), FAT32, NTFS, Unknown

Network Connections

Def	Column	Description
✓ 1	Name	name of connection, eg. Local Area Connection
✓ 2	Type	n/a (future feature?)
✓ 3	Status	n/a (future feature?)
✓ 4	Owner	n/a (future feature?)

My Network Places

Def	Column	Description
✓ 1	Name	name of tool or applet
✓ 2	Comments	description of tool or applet

Printers and Faxes

Def	Column	Description
✓ 1	Name	device or driver name (descriptive name, not file name)
✓ 2	Documents	number of documents in queue (ready for printing)
✓ 3	Status	Ready, Printing, error status, etc. - driver dependent
✓ 4	Comments	user assigned, or place there by installation
✓ 5	Location	user assigned? network?
✓ 6	Model	brand name, model name

Recycle Bin

Def	Column	Description
✓ 1	Name	file or folder name
✓ 2	Original Location	original file or folder location
✓ 3	Date Deleted	date the file or folder was deleted
✓ 4	Size	file size or folder contents size (if enabled)
✓ 5	Type	Windows associated descriptive file type
✓ 6	Date Modified	date of last file/folder modification

2.4.12 Autosize Columns

This menu item automatically resizes enabled columns to fit the data shown in the Files pane, including items which must be scrolled into view. Data items should fit exactly (no extra space) and should not be truncated. This operation does not affect other tabs, or newly opened tabs.

Before

After

Note: **Autosize Columns** is not a *one-time* option setting - you may need to use this feature on other folders.

2.4.13 Save Column Layout as Default

This function saves the current column layout as the **default** for any future new tabs, but only of the same *folder type*. The column layout saved includes:

• which columns are enabled,

- column order and

- column width.

Setting the default columns using the *Default Columns. . .* option, accomplishes nearly the same thing, but uses the current default column widths rather than the current column widths.

For more information about column settings, see *here*.

2.4.14 Change Display Window Colors. . .

This dialog allow you to customize the colors and font of the Display Window. See below for a brief *description* of the color gradient used as a background.
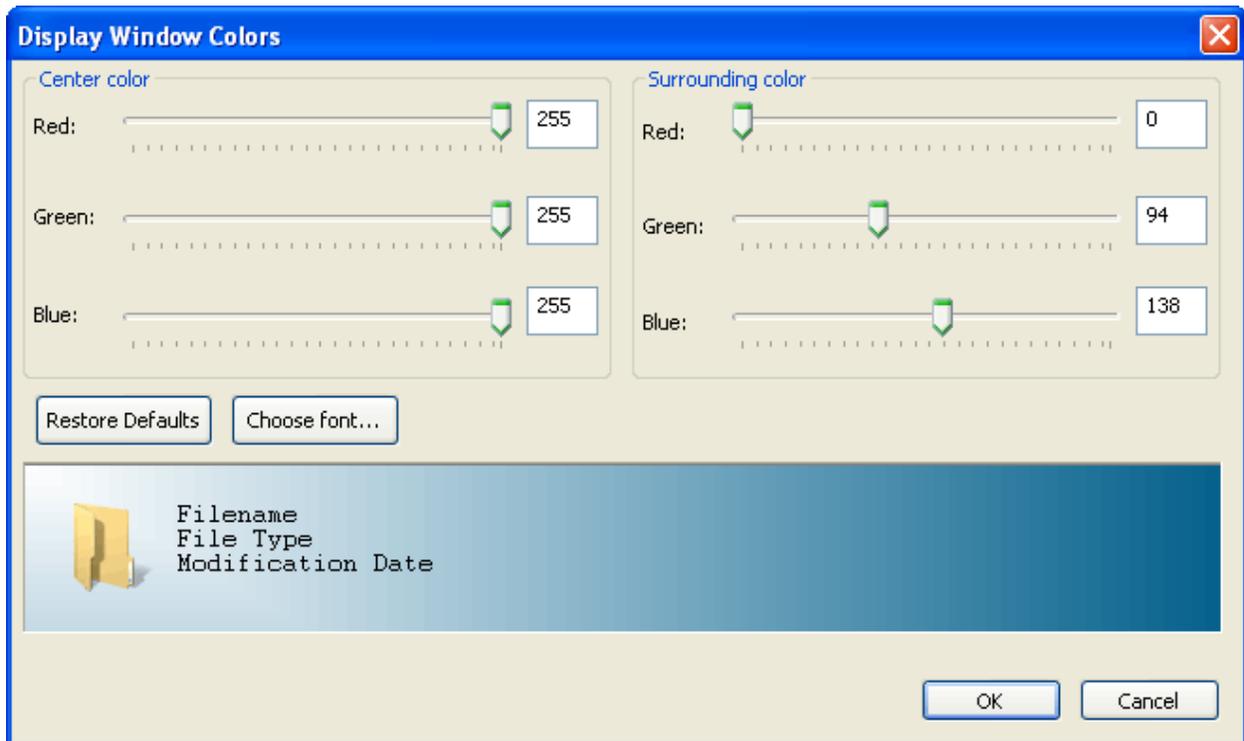
Description of color gradient

The Display Window background is a diagonal color gradient in 2 directions. Think of the **Center color** as the *starting* point of the gradient - a single point in the top-left corner of the **Display Window** ; the **Surrounding color** is the *final* color, located in the bottom-right corner. The gradient is *drawn diagonally* from top-left to bottom-right, but also in a pseudo-complementary (ie. at 90°) direction. The end result is a pleasing diagonal gradient with colors on the left generally close to the **Center color** and colors to the right generally close to the **Surrounding color**. The algorithm behind drawing the gradient is not useful; the colors should just be adjusted visually for the best combination.

Center color

The controls for the **Center color** (ie. top-left corner) allow RGB (Red/Green/Blue) adjustment, either by moving the slider or by entering an integer (0..255) in the edit box to the right of each slider. For a little help with RGB colors, see this site.

Surrounding color

The controls for the **Surrounding color** (ie. bottom-right corner) allow RGB (Red/Green/Blue) adjustment, in a manner similar to the *Center color*.

Restore Defaults

This button restores **Explorer++**'s built-in defaults for the Display Window, that is

- Center color = RGB (0,0,0)

- Surrounding color = RGB (0,94,138)

- Font = "Segoe UI"

    - height = 10 (approximate - device dependent)

    - weight = 500 (medium)

    - no italic

    - no underline

    - no strikeout

    - default character set

    - high quality output

    Note that the "Segoe UI" font is an included font in Windows Vista and 7; most Windows XP users will not have access to that font and their system will substitute an available font, probably "Courier". A similar font might be

    - Lucida Sans (Bold),

    - Microsoft Sans Serif (Bold),

    - Tahoma, or

    - Trebuchet MS

    but nearly any font may be used.

Choose font...

This button opens a standard Windows dialog allowing the user to choose a suitable font for text in the Display Window. Headings and information text both use this font. See comments in *Restore Defaults* regarding the font used.

Display Window preview

This area previews the color gradient and text using the selected *font*. Changes made to the *Center color* and *Surrounding color* are previewed, as the changes occur, so that the user may choose the color gradient visually.

2.4.15 Filter

Filtering allows certain files to be masked from view in the Files pane; **only** the files you specify (and folders) will be visible. This helps find files quickly, particularly in folders which contain a large number of files, such as the *Windows* folder. Filtering can be applied on a tab-by-tab basis.

Set Filter. . .

`Ctrl + Shift + F`
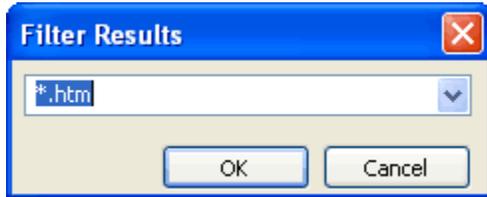
The filter specified is in the form of a wildcard pattern, in a manner similar to **Wildcard Select**. See *here* for a description of wildcard patterns. When the pattern is set (by clicking OK), then the filter is *applied* automatically to the current tab and only those files which are a match for the wildcard pattern are displayed.

The **Set Filter. . .** dialog *drop-down* shows previous wildcard patterns used in filtering.

Apply Filter

This menu item is a toggle which enables/disables filtering; filtering (when enabled) always uses the last set wildcard pattern, as shown on the **Set Filter. . .** (Filter Results) dialog.

When filtering is enabled, the following image is present in the center of the Files pane:

If filtering is disabled, the files which are now visible may be placed at the top of the file list; a *Refresh* may need to be done in order to sort the files.

2.5 Actions

2.5.1 New Folder

`Ctrl + N`

This menu item creates a new folder in the current folder (tab). The default name is **New Folder**, but creation of the folder automatically enters *rename* mode: so that the new folder may be renamed by the user to something more useful.

Creating a new folder inserts it into the file list (Files pane), obeying the setting of *Insert new items in their sorted positions*.

2.5.2 Split File...
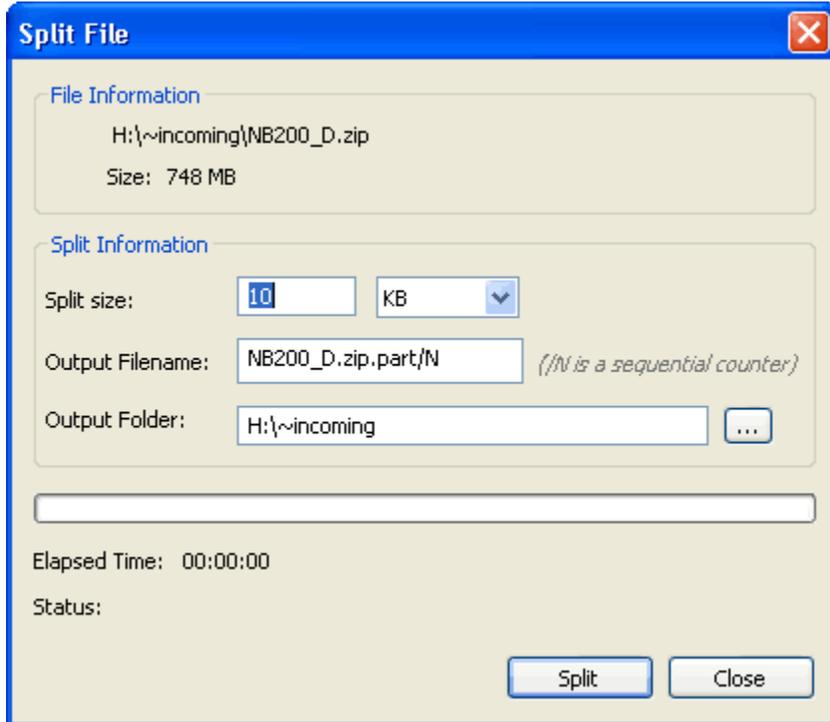
This function provides a useful operation - it splits the **selected** file into equal sized *sub-files*. While this may not seem that useful at first glance, many email providers, for example, have a limitation on the size of attachment that can be sent with a single message. If the file is over that size, then the server will automatically reject that message entirely; the file can be split, sent in parts, then *re-assembled* when received.

The companion function - *Merge Files...* - restores the original file by combining the *sub-files*.

Split size:

Enter the split size in the left box; set the units in the right box using the *pulldown*. Units are determined by the units *drop-down* list to the right and may be

- Bytes,
- KB kilobytes = 1024 (2^{10}) bytes,
- MB megabytes = 1024 KB or 1,048,576 (2^{20}) bytes, or
- GB gigabytes = 1024 MB or 1,073,741,824 (2^{30}) bytes.

The final size of the individual *sub-files* is a combination of the size and units selected.

Output Filename:

The output filename is actually a pattern for consecutive naming/numbering of the output files; the string "/N" (capital N only!) represents the *variable* numeric part of the future file names (e.g. 1, 2, 3, etc.). The default is <original filename>.part/N, which in the sample dialog above (original file = NB200_D.zip) would create *sub-files* of

- NB200_D.zip.part.1
- NB200_D.zip.part.2

- NB200_D.zip.part.3

- etc.

The *sub-file* naming pattern **must** contain a variable (i.e. /N) and the numeric variable portion of the new files always starts with 1.

Tip: Be careful with leading zeroes in the output filename; they are treated as constants. For example, using NB200_D.00/N as the pattern yields files NB200_D.001, NB200_D.002 but would also yield NB200_D.0010 instead of the *anticipated* NB200_D.010! **Explorer++**' s *Merge Files. . .* routine can handle this naming, but other *split/join* applications may not behave properly.

Output Folder:

The output folder (where the sub-files will be placed) defaults to the same folder as the target file. The folder may be chosen by typing (the target folder must exist) or by using the . . . button to the right which opens a standard *Browse for folder* dialog (see *here* for a picture of this Windows dialog).

Split

Clicking the Split button commences the operation, creating *sub-files* of the specified size. The last file may not be the same size; it is simply the remaining portion of the original file. Files are named the same as the target file, but with a number appended, that is, if **message.txt** is split, *sub-files* will be **message.txt.1**, **message.txt.2**, and so on.

Note: If **Split File. . .** involves creating a large number of files (e.g. > 500), there may be delays in response, depending on the speed of your computer. Be patient - **Explorer++ will** finish the operations properly.

2.5.3 Merge Files. . .
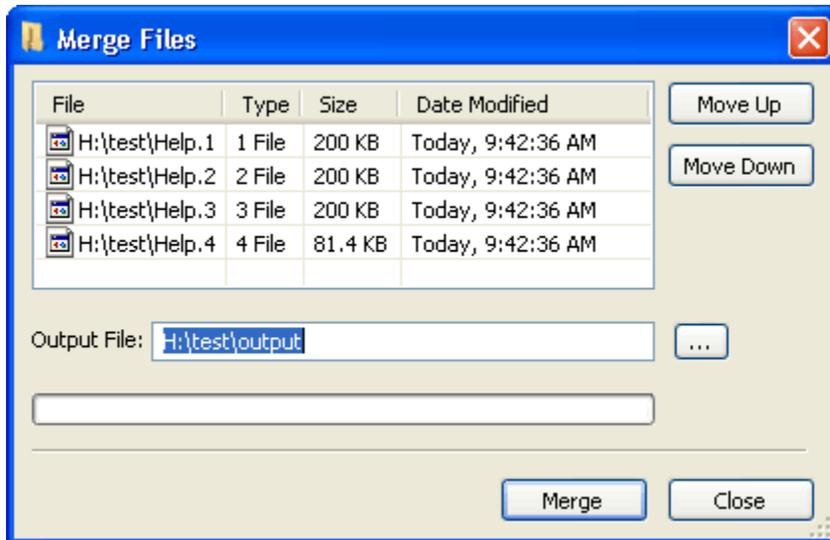
This function merges **selected** files (which were previously split by *Split File. . .* , or other utility) into a single file. The merge operation is a simple binary concatenation of the files, in the order shown.

Files to be merged

This area contains the list of files to be merged; the order in which they appear is the order in which they will be merged. The order may be changed through the use of the *buttons* to the right. There is a good likelihood that - if the files were created by **Explorer++**'s *Split File* routine, they will already be in the correct order. Note that the **Date Modified** column respects the setting of the *Show user friendly dates* option.

Move Up/Down buttons

These buttons help manipulate the file list by moving the selected file (one only!) up or down in the list. If the files were previously split by *Split File...*, they *should* already be in order.

Note: Multiple files may be selected by default, but **Up/Down** operations only operate on a **single** file.

Output File

This edit box contains the path and file name to the final *merged* output file. The default name is **output** (no extension); the file will be generated by default in the same folder as the files in the merge *list*.

Output Folder

This button opens a standard Windows *Browse for folder* dialog (see *here* for a picture of this Windows dialog), allowing the user to place the output file in a different location.

Tip: The *Browse for folder* dialog **only** selects a folder; the user must then add a file name. Failure to add a file name will result in an error.

Example: *Browse for folder* yields "F:\myfiles\temp",

but user must add "\final.zip" as filename, making

Output File: "F:\myfiles\temp\final.zip".

Merge (or OK)

Clicking **Merge** (or **OK** - button text may change) performs the file merge operation, creating a new file. The merge is done using simple binary concatenation without regard for contents. While any set of files can be merged, the **Merge Files** action is designed to *re-combine* files created by the *Split File* action.

Tip: Although individual files may be selected before merging (eg. hold **Ctrl** key and select random files), this has no effect. All files in the dialog will be merged in the order shown, regardless of any selection.


Errors

If a **Merge** error occurs, it is likely for one of the following reasons:

- the named output file already exists. Merge will not offer to overwrite the file.

- no file was specified (e.g. C:\files\temp\)

- the output file name matches a folder name, e.g. C:\files\temp where temp is a sub-folder. This may be a result of choosing an output folder but forgetting the file name - see *Output Folder:* above.

2.5.4 Destroy File(s)...
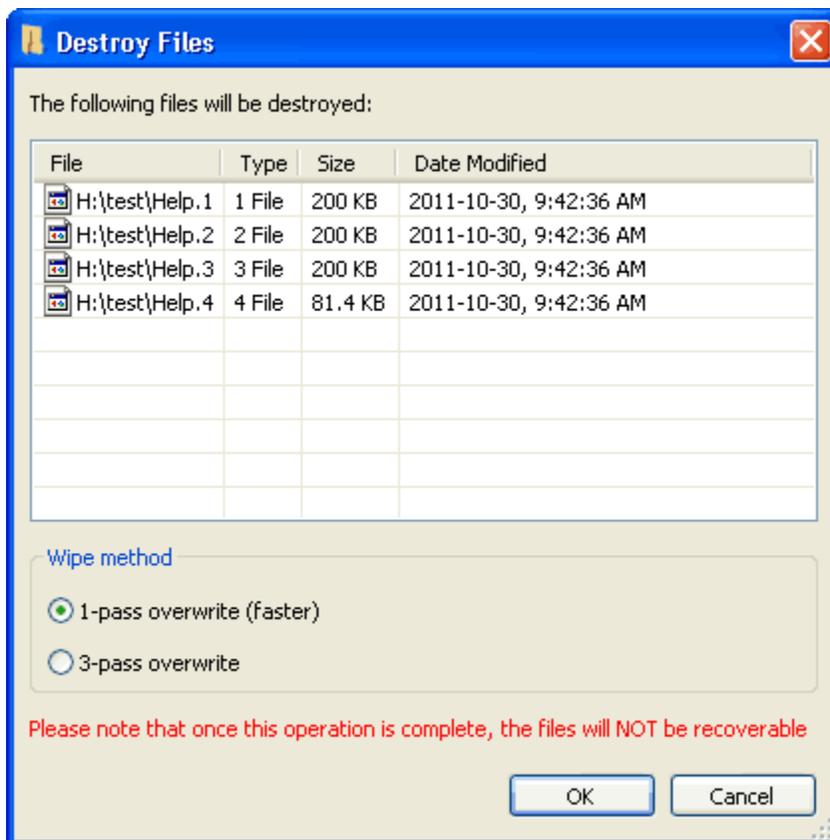
The **Destroy File(s)...** action securely deletes selected files by overwriting their file space, preventing file recovery. The dialog allows

- removal of files marked for deletion, and

- selection of either a 1-pass or 3-pass overwrite.

> **Caution:** The **Destroy File(s)** action is not the same as the *Delete Permanently* routine!

Why use secure deletion?

Today's computer stores **much** more than just games and documents; many users access secure websites using password protection to

- buy and sell stocks and securities,
- manipulate their bank accounts and move money electronically
- pay bills
- make purchases using credit card information.

The computer of such a user likely contains records of such transactions which, in the wrong hands, could lead to identity theft, or worse! Although encryption is often used as a security measure, it has already been shown that RSA public key encryption - once considered unbeatable (even using sophisticated computers) in less than thousands of years - may be hacked under certain conditions in as little as several days. In short, we must always think about

- what is possibly on our computer - passwords, sensitive data, etc.,
- and how can we protect it.

Most of us know that *deleted* files can be recovered fairly easily by tools such as Recuva, a freeware product. But today computer forensic tools can be used to read residual magnetism on hard drives, *possibly* re-constructing portions of data that *could* be used unlawfully. Secure deletion of files - in a way which renders them unrecoverable - is one simple way we can safeguard some of our personal data.

> **Note:** USB flash drives, although not magnetic, may still be recovered in a manner similar to hard drives. Secure deletion can be useful for flash drives as well.

What does Destroy File(s) actually do?

The **Destroy File(s)** - secure deletion - routine performs the following tasks on each file:

1. locates file data
2. determines the actual size of the file on disk, i.e. how many clusters are allocated to that file
3. opens the file, blocking any sharing mode, thus stopping the file from being opened (by another network user, for example) while it is being overwritten
4. extends the file out to the end of its last sector, eliminating file slack
5. starts at the beginning of the file, and writes in the first-pass data, 0x00 over the length of the whole file. This overwriting effectively eliminates the former contents of the file.
6. if **3-pass overwrite** has been chosen, then
   1. starts at the beginning of the file, and writes in the second-pass data - 0xFF - over the length of the whole file.
   2. then, writes in a *random* set of data, again byte by byte, as the third pass. This random data is obtained via the operating system's cryptography functions, and as such, is extremely difficult to duplicate.
7. flushes (ie. empties) all memory buffers connected to the file
8. closes the file and *frees* the Windows file handle used for the above procedures

---

9. deletes the file **without** sending it to the Recycle Bin.

Speed

Destroy File(s) writes data either once or three times, one byte at a time, and is, therefore, somewhat *disk intensive*. What this means is that the routine is not as fast as normal file writing. Calculation of an estimated write time would be difficult, since it would be dependent on a number of *uncertain* aspects, namely

- the type (number of cores, etc.) and speed of processor
- the type of drive (IDE, EIDE, SCSI, SATA, USB, flash, etc)
- fragmentation of the file - fragmented files contain clusters that may be physically farther apart
- amount of system memory - this affects the speed of system operations which indirectly could affect write times
- etc.

Attempting to use **Destroy File(s)** on a large file may create a **very** long wait time, so the following estimates are included here for reference and comparison:

Hardware: Pentium III processor, 1.2 Ghz, 768 Mb ram

Operating System: Windows XP SP2

Drives: 1: EIDE, 160 Gb; 2: USB, 1.0 Tb

	Files	**1**	**2**	**3**
	Sizes	1 Mb (1,048,576)	10 Mb (10,485,760)	100 Mb (104,857,600)
1-pass	Drive 1	± 2.1 sec	± 19.6 sec	± 3.2 min
	Drive 2	± 1.8 sec	± 16.0 sec	± 2.6 min
3-pass	Drive 1	± 9.9 sec	± 1.6 min	± 16.0 min
	Drive 2	± 8.5 sec	± 1.4 min	± 14.0 min

The times for the USB drive are shorter, I believe, owing to the larger drive capacity (shorter seek time). To put these times into perspective, for the listed hardware, one could estimate the secure deletion times as

- 2 seconds/megabyte for 1-pass, and
- 10 seconds/megabyte for 3-pass.

This means that deleting a DVD image file (± 4 Gb), for example, using 3-pass secure deletion, could take as long as 11.5 hours! Sleep on it!

Early termination

On the *hardware* listed above, the Windows Task Manager reports that **Explorer++** is using between 95% and 99% of the CPU time during a **Destroy File(s)** operation, despite the fact that the process is operating at *Normal* priority. The effect of this is that

- the appearance of **Explorer++** (not the **Destroy File(s)** dialog) is that of a program that has "stopped responding to the system", and
- attempting to switch to another application (via the taskbar) and do something else is a waste of time.

If **Destroy File(s)** is launched in error on a large file (see *times* above), it can be prematurely terminated using the Task Manager by

1. opening the Task Manager using **Ctrl+Alt+Delete**

2. selecting the process **Explorer++.exe**

3. clicking *End Process*

4. agreeing to the warning dialog

The effect of this is to forcefully close **Explorer++** and, of course, the **Destroy File(s)** dialog (a child window), freeing up your system resources. However, take note that

- settings for **Explorer++** (ie. open tabs, views, etc.) may not have been saved, and

- the file(s) targeted for secure deletion were, in all likelihood, damaged (partially overwritten), but not fully deleted.

Disclaimer

Although the **Destroy File(s)** action is deemed to be secure, **Explorer++** is released under the GNU General Public License and no warranty as to suitability is offered. Section 15 of the GNU license states that the program is provided "as-is" without warranty of any kind.

In other words, if you use the **Destroy File(s)** routine, and you **still** get busted for importing ivory when the police seize your computer and find your files, you're on your own!

2.6 Go

The top portion of the **Go** menu allows quick access to special locations, providing access to data which the user may require. The bottom portion of the menu allows quick access to special locations, for computer configuration, etc.

2.6.1 Back

`Alt + ←`

Explorer++ maintains a list of previous locations, or history, for each tab. The **Back** (and *Forward*) items on the **Go** menu can walk through this history, much like hyperlinks in HTML browsers (eg. Mozilla Firefox).

Back returns the currently active tab to an older location, closest to the current location in the history. If this item is disabled, the Files pane (active tab) is displaying the oldest location in the history, the location when the tab was first created, or when **Explorer++** was first opened.

History

A tab's history can be thought of as a collection (or list) of locations in the order they were visited, with the oldest at one end, and newest at the other. Locations are added to history each time a location is visited by

- clicking on a drive in the *Drives toolbar*,

- clicking on a folder (or other item) in the Folders pane,

- double-clicking (or single click, with *Single click option* enabled) on a folder in the Files pane,

- clicking on a Bookmark, either from the *Bookmarks menu* or the *Bookmarks toolbar*,

- clicking on an item from the *Go menu*, **except** Back or Forward,

- clicking on the Up button on the *Main toolbar* (same function as on the Go menu), or

- entering an address in the *Address bar* which changes the location.

Executing **Back** or **Forward** simply walks the Files pane through the current history for a tab; the functions do not add to, or subtract from, the location history. It is possible that history locations may be duplicated, if they were visited by one of the above methods more than once. As **Back** or **Forward** is used (either from the **Go** menu or the *Main toolbar*), items in history are *re-distributed* between the *Back history* and *Forward history*; Explorer++ maintains histories in both directions.

2.6.2 Forward

`Alt +` →

Explorer++ maintains a list of previous locations, or history, for each tab. The **Forward** (and *Back*) items on the **Go** menu can walk through this history, much like hyperlinks in HTML browsers (eg. Mozilla Firefox).

Forward returns the currently active tab to a newer location, closest to the current location in the history. If this item is disabled, the Files pane (active tab) is displaying the newest location in the history.

See *here* for a more detailed explanation of a tab's history.

2.6.3 Up

This menu item changes the address of the currently active tab to the *parent* folder (or location), as shown in the Folders pane (providing it is up to date).

The parent folder (or location) is the folder which contains the folder currently displayed.

If Up One Level is disabled, the Files pane is currently displaying the highest level available in the Folders pane (*virtual* tree) - *Desktop*.

2.6.4 My Computer

This is a *virtual folder* which displays

- all *volumes* on the computer. A volume is a single storage location, usually with a single, drive letter unique to the computer. It can be an entire hard (or floppy) drive, a partition within a drive, a removable (eg. USB) drive, an optical drive (ie. CD, DVD or *Blu-Ray Disc*™, or a "drive" (with a drive letter) assigned to a folder in the NTFS file system.

- *virtual folders*, such as *Control Panel*

- renamed real locations such as Bob's Documents (probably *C:\Documents and Settings\Bob\My Documents*) and Shared Documents (probably *C:\Documents and Settings\All Users\Documents*); drive letters may vary with different installations of Windows.

Note that **My Computer** may contain different items, or may display differently in each version of Windows.

2.6.5 My Documents

This is a real folder (probably *C:\Documents and Settings\<username>\My Documents* - drive letters may vary) which Windows creates as part of its installation. The purpose for this folder, at least as intended by Windows, is to act as a parent folder for all the user's documents and data. This may not suit each user's purposes, but many programs default to this location for data storage.

2.6.6 My Music

This is a real folder (probably *C:\Documents and Settings\<username>\My Documents\My Music* - drive letters may vary) which Windows creates as a subfolder of *My Documents*, intended to house digital music files (eg. MP3, WAV, etc.).

2.6.7 My Pictures

This is a real folder (probably *C:\Documents and Settings\<username>\My Documents\My Pictures* - drive letters may vary) which Windows creates as a subfolder of *My Documents*, intended to house digital image files (eg. JPG, GIF, PNG, etc.).

2.6.8 Desktop

Desktop is a *virtual folder*, considered by Windows to be the parent (ie. the topmost item in the hierarchy) of all items in the *virtual tree* shown in the Folders pane. The practical use for **Desktop** is that it always contains all items shown on the user's actual desktop (ie. the Windows screen when no application windows are visible).

Although **Desktop** is considered to be a virtual folder (for purposes of the computer's *virtual tree*), it is really comprised of 2 real folders

- *C:\Documents and Settings\<username>\Desktop*, and
- *C:\Documents and Settings\All Users\Desktop*

The (root) drive letter may vary with different installations of Windows.

2.6.9 Recycle Bin

This is a *virtual folder* that displays deleted files and folders from various locations. This is the same **Recycle Bin** usually shown on your *Desktop*. Selected items may be restored by using the "Restore" menu item on the context (ie. right-click) menu. Note that files deleted from a **Local** USB (ie. external) drive will not be visible in **Recycle Bin** unless the drive is physically and logically connected to your system. Files deleted from a Removable USB drive (eg. *flash* or *jump* drive) do not appear in the **Recycle Bin**.

2.6.10 Control Panel

This is a *virtual folder* which displays various tools (ie. *applets*, or small applications) used to configure your system or other applications. Most *applets* are physically housed in the WINDOWS\system32 folder, but some may be elsewhere. This is the same **Control Panel** usually shown on your *Start Menu*.

2.6.11 Printers and Faxes

This is a *virtual folder* which displays *links* to all printers and faxes connected to your system. This location may also show drivers which control various devices. The context (ie. right-click) menu for each device or driver may show various control items including *Properties*, which usually allows some setup procedures.

2.6.12 CD Burning

Windows XP and higher (Vista, 7) has built-in routines for native *burning* (ie. writing) of CDs. This burning requires a buffer where data is kept that will be written to the CD; this buffer location is a folder reserved especially for this purpose - "**CD Burning**". Files and folders contained in this folder are slated for writing to CD, but could be removed (ie. deleted) if necessary prior to the actual write.

2.6.13 My Network Places

This is a *virtual folder* which displays frequently visited network locations which have been added; these locations are essentially "network bookmarks". These network places may be added using the **Add Network Place** wizard, displayed in the Files pane. Two other network *tasks* (wizards) are also available from the Files pane - the **Network Setup Wizard**, and the **Wireless Network Setup Wizard**. **Explorer++** displays these three wizards in the Files pane along with previously set up network places, while Windows Explorer shows them (under expanded titles) as tasks, when the Folders pane is turned off.

Incomplete

My Network Places currently does not provide access to the actual network, except through the Folders (*treeview*) pane. Network places are housed under the *Entire Network\Microsoft Windows Network* item.

2.6.14 Network Connections

This is a *virtual folder* which displays networks which your computer is connected to, or *can* connect to.

Incomplete

Network Connections currently does not provide information about the *Type*, *Status* or *Owner* (available *columns* for display). This information (and more) can be viewed by selecting **Control Panel** in the Folders pane, right clicking **Network Connections** in the Files pane (for the context menu), and selecting **Explore** - this opens Windows Explorer which will display this information.
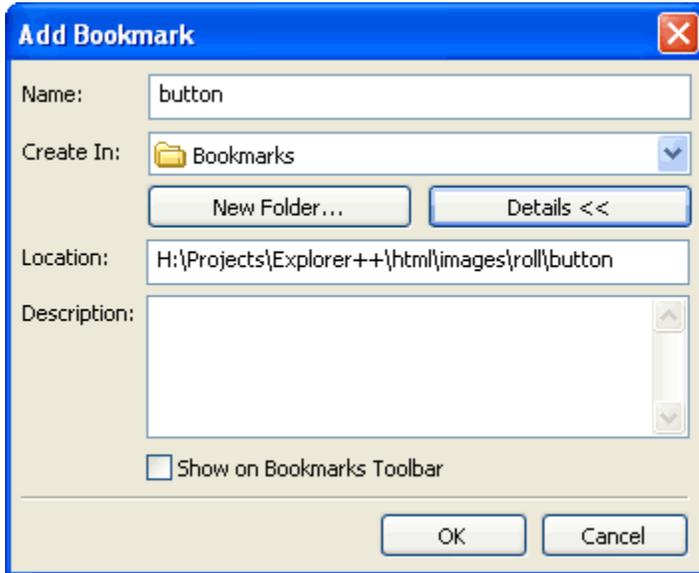

2.7 Bookmarks
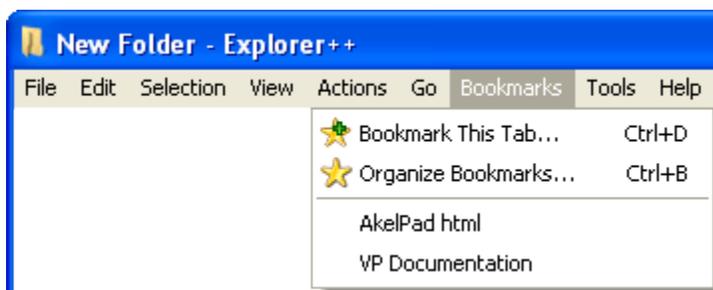
User bookmarks are displayed at the bottom of the menu. Clicking on a user bookmark "goes there", ie. set the active tab in the Files pane to the location for that bookmark. If the location does not exist (ie. folder for bookmark has been renamed or deleted), no change is made.
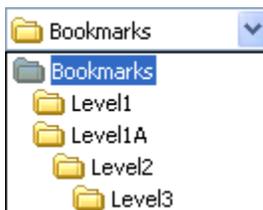
2.7.1 Bookmark This Tab...

`Ctrl + D`

Name:

The **Name** field allows entry of the final bookmark name, as it will be displayed on the *Bookmarks menu*. You should choose a name that conveys (to you) the usage of this location.

Create In:

Bookmarks may be built in a hierarchy, similar to the tree shown in the Folders pane, through use of the *New Folder...* button. Bookmarks, therefore, may be created on any level (ie. in any subfolder) which exists. Clicking this field (or the *drop-down* button) opens a list which shows all current levels; the user may select which folder or level to create the new bookmark in, as in the image.

New Folder...

This dialog creates a subfolder in **Bookmarks** (the *root* level for user bookmarks), or within any other bookmark subfolder which currently exists. Note that folders may **contain** bookmarks, but do not have a bookmark associated with them.

Bug

Creation of a new folder (ie. one level below Bookmarks) is fine, but creation of second folder within the first (ie. two levels down) causes the folder names to be displayed incorrectly on the **Bookmarks** menu. All subfolders past the first level are shown with the same name - the name of the lowest level. **Workaround:** The *Organize Bookmarks* dialog displays the bookmark hierarchy and folder names correctly.

During creation of a new folder on the same level, **Explorer++** does not check for duplicate names.

See also - fatal error (bug), when *deleting* empty bookmark folders.

Location:

Normally, this field shows the URL (ie. location) of the currently active folder in the Files pane, but you may edit it as desired. Make sure that the location you edit actually exists.

Note: Any bookmark with no location (ie. empty string) will open in the *default new tab folder*.

Description:

This field is normally not shown, but appears as part of the bookmark's *Properties* dialog in the *Organize Bookmarks* tool. Any user comments about this bookmark may be entered here, such as, perhaps:

- the use of this folder (eg. "Temporary storage for untested software", "Killer apps!")
- a date ("Wednesday's accounting updates")
- backup information ("These ZIP files have been archived")
- etc.

Show on Bookmarks Toolbar

Checking this box (or menu item) forces the bookmark created to be shown on the *Bookmarks toolbar*. If this box is not checked at the time of bookmark creation, it may be checked later as part of the *Properties* dialog in the *Organize Bookmarks* tool. Folders housing bookmarks may also be designated to show on the Bookmarks toolbar.

2.7.2 Organize Bookmarks. . .

`Ctrl + B`
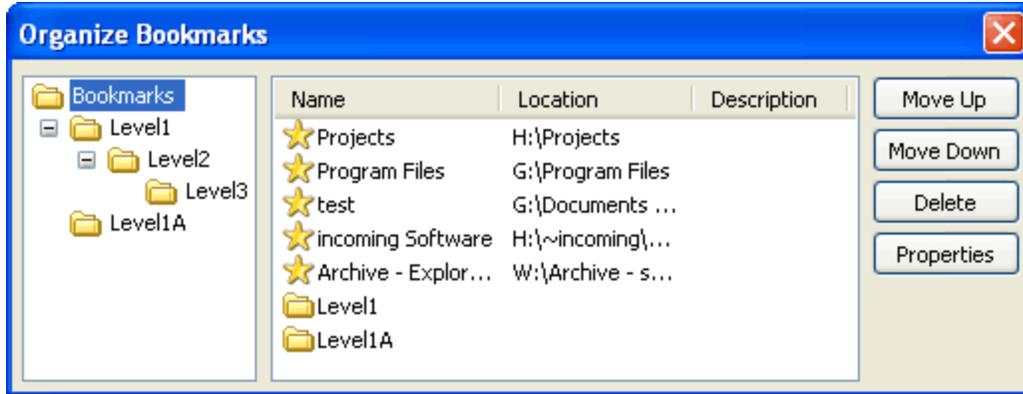
Folders tree

This pane shows the hierarchical *tree* structure of the bookmarks collection. Folders can be expanded to show their contents (more folders only). Selecting a particular folder displays the contents (including bookmarks) in the **Bookmarks & Folders** pane to the right. The folder **Bookmarks** is the parent (or root) folder of all subfolders or bookmarks.

Bookmarks & Folders

This pane displays the contents of a selected folder in the **Folders tree** to the left. In addition to the bookmark or folder's name, it's location and any comment (Description) are displayed. The three columns may be resized (by dragging the edges of the column headers), but not dragged into a different order.

Double-clicking on a bookmark (⭐) transfers the current tab to that location, ie. *goes* to the bookmark, while double-clicking on a folder (📁) opens that folder and displays its contents.

Context Menu
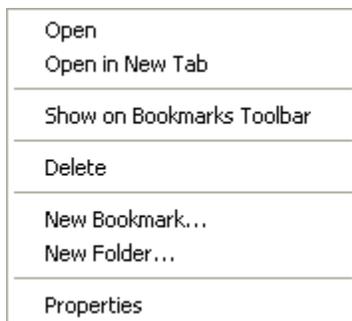
Each bookmark item in this pane has a context (ie. right-click) menu which is quite similar to the context menu for items on the *Bookmarks toolbar*, with the addition of a *Show on Bookmarks Toolbar* item.

Incomplete

The *New Bookmark. . .* and *New Folder. . .* items on this menu are not yet coded, ie. they are non-functional.

Move Up, Move Down

These buttons move a selected bookmark (or folder) in the Bookmarks & Folders pane up or down in the bookmarks list. Only a single bookmark or folder may be selected at a time.

Delete

This button deletes (with a confirmation dialog) a single bookmark or folder selected in the Bookmarks & Folders pane. Deleting a folder obviously deletes all the bookmarks and folders it contains.
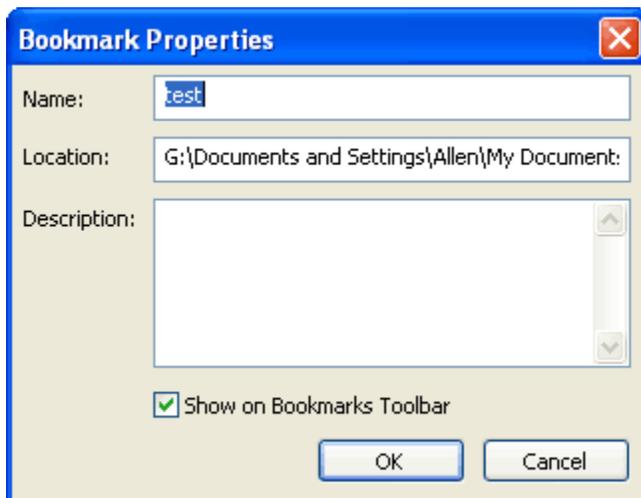

Bug

Deleting a folder does not update the Folders tree until the next opening of the **Organize Bookmarks** dialog.

It is possible to create a new folder, then save the bookmark elsewhere or cancel saving the bookmark, thereby creating an empty folder. Attempting to subsequently delete an empty folder will cause a **fatal crash**.

Workaround: Prior to deleting a folder, select it in the Folders tree to ensure that it is not empty. If necessary, add a dummy bookmark to the folder prior to deleting it.

Properties

You edit the properties for any item selected in the Bookmarks & Folders pane. If the item is a folder, then the **Location:** field is absent.

2.8 Tools

2.8.1 Search...

`Ctrl + F`

`F3`

more room in the edit box.

Directory

This field should contain the folder in which to perform the search, or the parent folder, if the *Search Subfolders* box is checked. The button beside the field opens a Windows *Browse For Folder* dialog (see *here* for a picture of this Windows dialog), allowing the user to navigate to and set the **Search Directory**. A *drop-down* list is available with quick access to previous search directories.

The **Search Directory** defaults to the folder/directory of the currently active tab.

Attributes

Checking any *attributes* provides a filtering mechanism; files searched must satisfy the Filename *pattern* AND the attributes if they are to be included in the results.

Case Insensitive

The state of this checkbox determines whether text case (i.e. *UPPER* or *lower* case) will be considered during the search.

   • **checked** - searches are case insensitive, that is, *UPPER* and *lower* case are considered the same. Searching for "gif", for example, will also find "GIF" and "Gif".

   • **not checked** - searches are case sensitive and the target and results case must match exactly, that is, "gif" "GIF" "Gif".

Search Subfolders

Normally, the search would be done only in the scope of the folder shown as the *Directory*. However, if this box is checked, all subfolders (and subfolders of subfolders, etc.) will be included in the search. Results can be from either the search Directory, or any subfolder.

Use Regular Expressions

If this box is checked, then the text entered in the *Filename* edit box is assumed to be a regular expression and the search is done using pattern matching instead of simple file name comparisons. Regular expressions, while appearing complicated to the user of text and wildcard expressions (e.g. "*.jpg"), can be a more flexible and powerful approach to searching for files and folders. Having this tool available - as well as *conventional* (i.e. non regular expression) text searching - provides superior capabilities and choice to the user. For help on regular expressions, see this MSDN page. Regular Expression searches obey the settings of the **Attributes** and **Search type** checkboxes.

Results pane

When the search is completed, this list shows the name and path (ie. location) of all files found which matched the criteria for (partial) name and attributes. The **Name** and **Path** columns are resizable, and *sortable* (ascending or descending, by clicking on the column header) for convenience when numerous results are visible.

Tip: Each file in the Results pane has a context (ie. right-click) menu. Two items on this menu are of particular interest:

- **Open** - this opens the file in any application registered to that file type or presents a dialog if the type is not registered. This may also be accomplished by double-clicking on the file.

- **Open file location** - this opens a new tab at the folder which houses the file. The target file is already selected in the new tab. If multiple files were selected originally, only a single tab is opened and a single file selected.

This context menu is constructed from Windows *shell* associations; items present will vary with the file type.

Note: The Microsoft .NET framework, when present on a system, creates a subfolder in the Windows folder called "*assembly*". The nature of the .NET installation may interfere with **Explorer++** (Windows Explorer's exploration is *modified*) successfully browsing this folder. If this folder is included in your search (usually by virtue of using the *Search Subfolders* option), some results unusual results (eg. incorrect file names, locations) may be expected, but these erroneous results are usually obvious.

Status:

This area shows

- the folder being searched (quickly!), as the search progresses,

- "Cancelled." - if the user stopped the search prematurely (by clicking the Stop button or by pressing **Enter** or **spacebar**), or

- the final count of folders and files found, when the search completes.

2.8.2 Customize Colors. . .

This tool allows **Explorer++** to display files in different colors, according to *rules* which modify the display of files by filename and/or attributes. Like the *Filter* routine, coloring certain files or folders aids in quickly finding what you want.
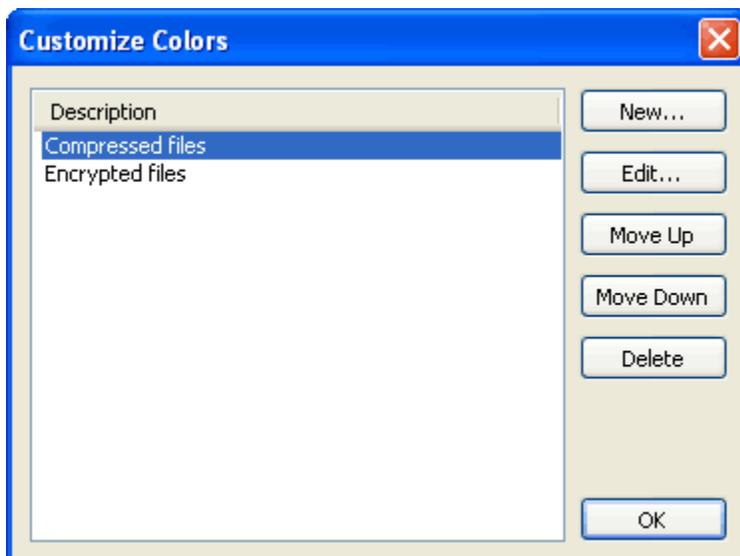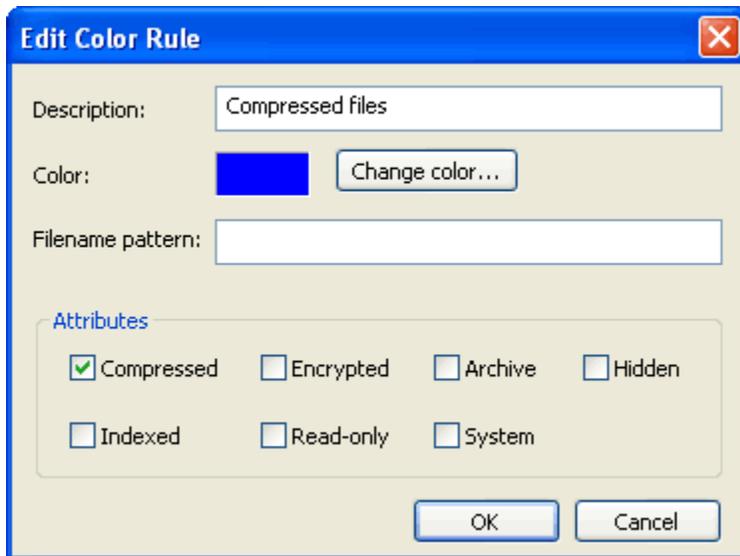
Rules list

This list shows all color rules, by their description (ie. *name*). Their application to files in the Files pane' is by their order of appearance in this list; the first rule (top) is applied first, the bottom of the list is applied last. Individual rules (ie. one at a time) may be selected for re-ordering, editing or deletion.

New. . .

This opens the **New Color Rule** dialog for creation of a new rule. The dialog is virtually identical in form and function to the *Edit Color Rule* dialog, buts adds to the rule list rather than editing an existing rule.

Edit. . .

This function edits the currently selected rule.

Description

Enter a description in this field; this is the name that will identify the rule in the *Rules list*.

Color

The color assigned to files which match this rule is shown in a small *swatch*.

Change color. . .

This button opens a standard Windows Color dialog which allows you to set a new color. You may

- choose a color from some common colors, or
- define one using
    - RGB (Red/Green/Blue) notation

  – Hue, Saturation and Luminescence values, or

  – visual selection

Filename pattern

Rules may specify a filename pattern in addition to, or instead of, *attributes*. If a file name matches this pattern, then that portion of the rule is satisfied. Filename patterns are entered using *wildcard patterns*, in a manner similar to the old DOS wildcards. For example, the filename pattern

_.*

would match any file containing an underscore character. The most common usage would likely be to match certain files by file extension, as in a pattern of

*.gif

which would match any GIF image files.

If attributes are also checked, then both the filename pattern AND the attributes must match in order for the rule to succeed and the file be colored.

Attributes

File **attributes** may be specified in addition to, or instead of, a *Filename pattern*. Attributes may be the *common attributes* (as can be changed from the File menu), or may also be NTFS specific, namely

  • **Compressed** - file is compressed using native Windows compression, or

  • **Encrypted** - file is encrypted using native Windows encryption.

Note that checking any attribute box in this dialog does not affect the attributes for any file, it merely reads the attributes as part of this rule.

Move Up/Down

The **Move Up** and **Move Down** buttons move a selected rule up or down in the list. Priority of application of rules to the Files pane is determined by their order in the list.

Delete

Clicking the **Delete** button deletes a rule (currently selected) from the list; a confirmation dialog is issued prior to deletion. There is no way to recover a deleted rule.

2.8.3 General Options

General
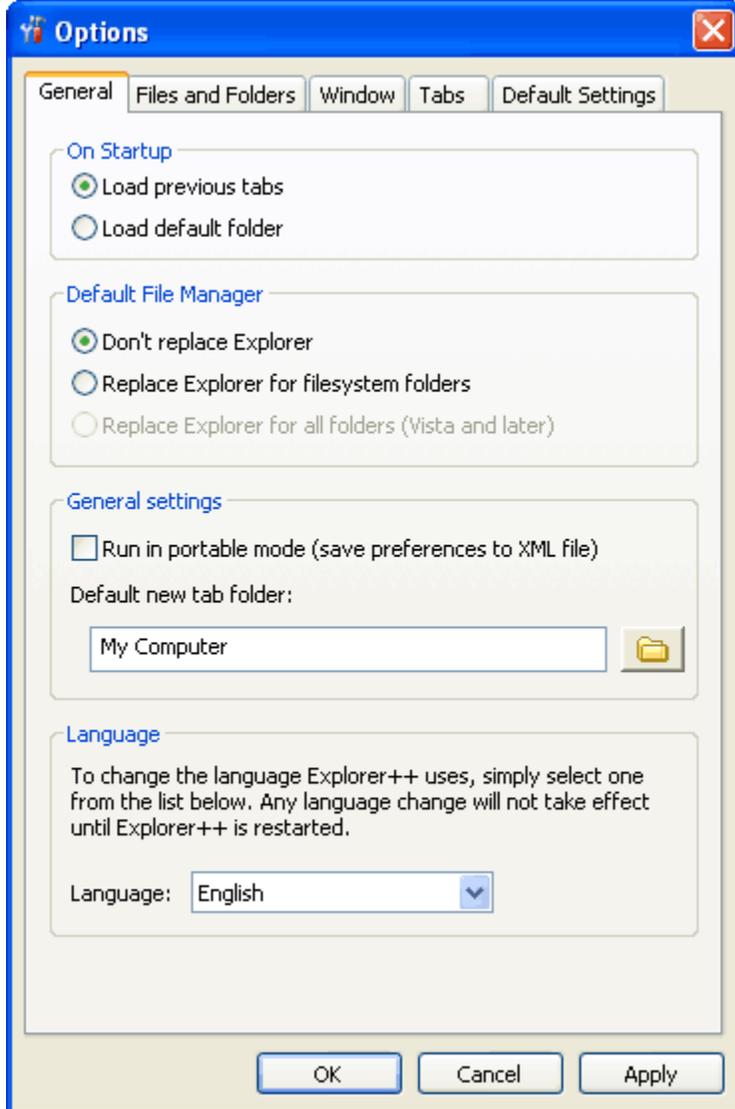
On Startup

System Default:	Load previous tabs

This option specifies the action **Explorer++** should take on startup. Two choices are available:

- **Load previous tabs** - This option forces **Explorer++** to save the current tabs and settings - either in the Windows registry, or in the *portable* XML file - and restore them on startup. This ensures that your *Tab bar* (and the settings for each tab) are "just the way you left them".

- **Load default folder** - This option tells **Explorer++** to just open the *default (new) tab* at startup - **no** previous tabs will be loaded (but see Note and Tip below!).

The Info and Tip following are applicable to the current version (v1.35) - future versions may vary from this behavior.

Note: **Explorer++** saves any tabs and settings from memory variables when the application is closed; changing from the **previous tabs** setting to **default folder** in the Options dialog only changes the setting. At the next startup, however, the memory variables will contain only the tab information specified by the setting - the previous tab settings are **still** in the registry or XML file, at least until **Explorer++** is closed.

Tip: If you have closed **Explorer++** and changed the setting to **Load default folder** (by accident?), you may still be able to restore your tabs for the next startup! One of the following changes (depending on whether you are using **Explorer++** in *portable mode* or not) must be made before the next session:

Registry: The key

```
HKEY_CURRENT_USER\Software\Explorer++\Settings\StartupMode
```

must be set to 1 (restore tabs).

XML file: The setting

```
<Setting name="StartupMode">2</Setting>
```

must be set to 1 (2=Load default folder). Settings in the XML file should be in alphabetical order; the StartupMode setting occurs around line 59. You can edit the file with a text editor like NotePad (not a word processor!).

The usual *caveats* apply for Windows registry editing - don't attempt this if you are unsure.

Run in portable mode

System Default:	No (if no config.xml file present)

Enabling this option (ie. checking the box) tells **Explorer++** to save and fetch all its settings in an XML (**config.xml**) file, located in the installation folder, usually …\Program Files\Explorer++ v1.2 (your version might be different). An XML file is a text file, structured hierarchically to enable **Explorer++** to re-construct its settings, etc. at program startup.

Note: At startup, **Explorer++** always looks for a valid XML file in the correct location, and if it is found loads it and uses those settings. The actual setting value for portable mode **is not saved**, either in the Windows registry, or in the XML.

If **Explorer++** is currently using portable mode (ie. XML file), and it is desired to switch back to using the Windows registry, the XML file must be either

- deleted, or
- renamed (eg. config.xml.bak)

and the portable mode option must be disabled (ie. clear the checkbox) prior to closing **Explorer++**, to avoid re-writing the XML file. Note that **Explorer++** creates the XML file (if this setting is enabled) immediately upon closing the options dialog.

Default new tab folder

System Default:	My Computer

This option allows the user to select a default folder to be used when a *new tab* is opened. The option presents a typical Windows *Browse For Folder* dialog, allowing the user to choose a folder. Prior to using this option, the default folder is *My Computer*.

If a folder is set by this option, then subsequently is renamed or deleted (ie. folder cannot be found), new tabs default back to *My Computer*.

Tip: The *Browse For Folder* dialog is a native Windows dialog and, as such, respects the settings from the **Control Panel/Folder Options** - **View** tab. Specifically, the

- *Do not show hidden files and folders,* and the
- *Hide protected operating system files (Recommended)* setting

if enabled will prevent display of some folders by this dialog. This behavior exists even if **Explorer++** is set to *display hidden* items, since **Explorer++** has no control over this dialog.

It is recommended that both of the above settings be changed to display all files and folders when using **Explorer++**, and to use its own control of these attributes.

Language

System Default:	English (9) - internal
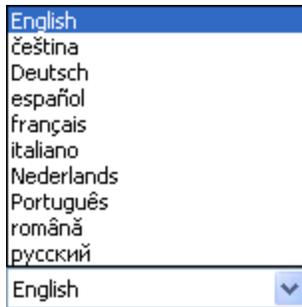
Explorer++ is a native English language application, but International users have contributed translation DLLs which, when placed in the *installation folder*, present the menus, etc. in a different language. The DLLs are not shipped with **Explorer++**, but may be downloaded from the translation page of **Explorer++**'s site at

https://www.explorerplusplus.com/translations

Once the DLLs are placed in the **Explorer++** installation folder, the languages will appear as choices on the language drop-down (or drop-*up*!) control for this option.

Note: **Japanese**, **Korean** and **Chinese** require support for East Asian languages to be installed. This installation is available from the Control Panel *Regional and Language Options* applet.

Note: **Sinhala** requires installation of either the SinhalaTamil Kit (Windows XP) or possibly a Sinhala LIP (Language Interface Pack - Windows Vista/7, untested), available from Microsoft Sri Lanka.

The translations are incomplete, but are an excellent start to the *globalization* of **Explorer++**.

To check the status of a translation, update a translation or submit a new translation, please see this project's page on Crowdin.

Files and Folders
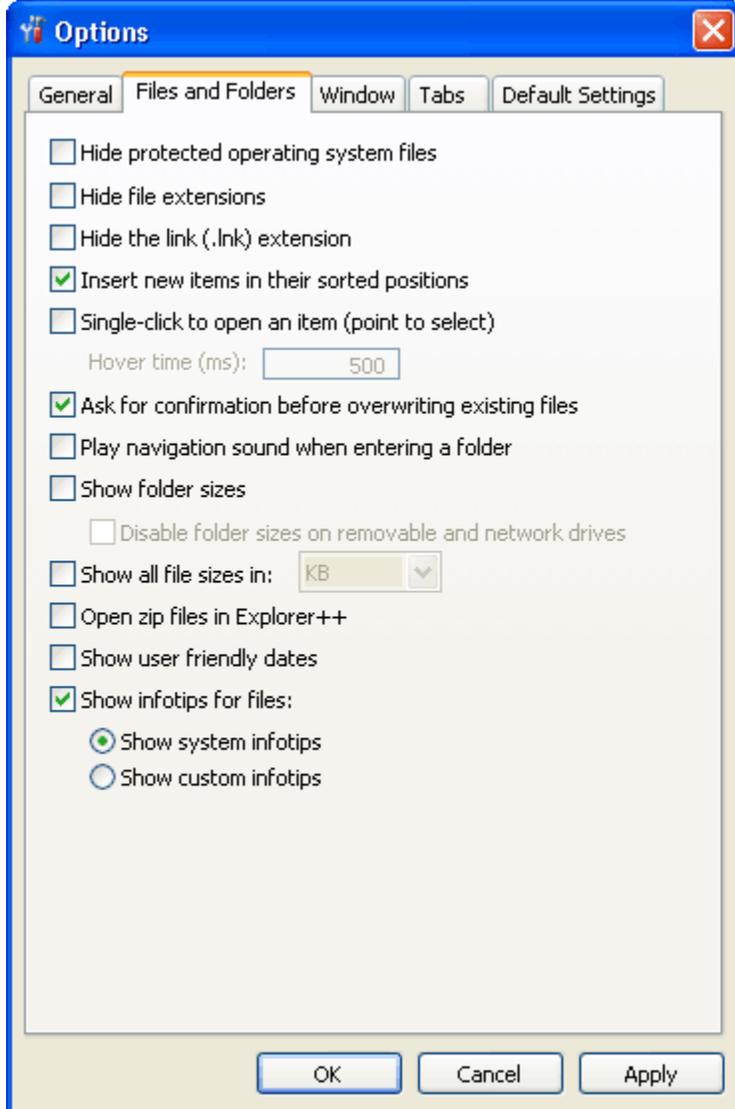
Insert new items in their sorted positions

System Default:	Yes

Explorer++ sorts items in the current Files pane (ie. tab) when a *Refresh* is done, or when refreshing is done automatically by changing the column sorting, etc. When new items are added to the tab by drag-and-drop, or by using the **New** item of the context (right-click) menu, newly added items are (by default) inserted in the correct location - folders at the top (or bottom, depending on the column header status), files in position by alphabetic order. Disabling this option, however, causes all new items to be placed at the bottom the of current list. This may be of benefit when editing folders, etc. and new files/folders can be determined at a glance.

Of course, automatic refresh (and sorting) is still done when column sorting is changed and at startup, etc.

Single-click to open an item

System Default:	No

Windows' default mode of operation (in most situations) is

- single-click on an item to select it (give it *focus*)

- double-click to activate it or execute it

The **X Window System** graphical user interface, popular with Unix/Linux-type operating systems, uses a different approach

- hover over an item to select it (give it *focus*)

- click once to activate it or execute it

Enabling this option causes **Explorer++** to behave like an X Window application - in the Files pane only. Some users may find this more comfortable (less mouse clicks) or faster.

Hover time (ms)

Hover time is the time taken for the item to be selected when using single click to open an item. Hovering over a item in single click to open mode is equivilent to clicking on it in double click to open mode.

Note: This behavior is available in Windows as a *system-wide* setting. The **General** page of the *Folder Options* Control Panel applet provides this functionality. **Explorer++**, as it is not a native Windows component, does not obey this setting.

Show folder sizes

System Default:	No

Checking this box forces **Explorer++** to calculate and display folder sizes in the Files pane. The feature might be useful if, for example, you were *house-cleaning* your drive and wanted to merge smaller folders with larger ones; folder sizes would be listed is the **Size** column, if *enabled*.

Caution: Although it sounds obvious, when this option is enabled, **Explorer++** calculates folder sizes - **always**. If you are *walking* through your folder tree to get to a particular location, large folders in the files pane will be calculated, even if they are only momentarily displayed. Windows is a multi-tasking, multi-threaded operating system; calculations continue even though the folder is no longer displayed. This may **greatly** impact your system's performance - use this option with care.

Disable folder sizes on removable and network drives

When folder size calculation and display is enabled, you may check this box to turn off the feature for removable and network drives, as a time-saving convenience. For network drives, this is a safety feature, as the system may not know ahead of time just how large the network drive is; time will be saved. In addition, network and removable drives are transitory - it may not be useful to know the sizes since tomorrow, the drives may not be connected.

Show all file sizes in:

System Default:	Flexible, Bytes if option enabled

When this box is cleared, drive and file sizes (and *folder sizes*, if enabled) are displayed in a *flexible* format, that is, in bytes, kilobytes (KB - 10^3 bytes), megabytes (MB - 10^6 bytes), etc., depending on file size.

Checking the box allows the user to choose from a variety of units, from bytes up to terabytes (10^{12} bytes) and even petabytes (10^{15} bytes)! File sizes will be displayed in the chosen units, even if files are small. This might prove useful in viewing the sizes of larger drives, etc. or media files (DVD files could be in the gigabyte (10^9 bytes) range.

Show user friendly dates

System Default:	Yes

Setting this option presents the

- Date Created,
- Date Modified and
- Date Accessed

columns in the Files pane with *user friendly* dates, such as "Today, *time*" and 'Yesterday, *time*" instead of the usual format. The usual format is determined by your system settings for Date and Time display and can be modified in the **Regional and Language Options** Control Panel applet.

You may open the Control Panel applet here: to customize your settings (Short Date format, Time).

Incomplete

To date, only "Today" and "Yesterday" have been implemented as user friendly dates.

Window
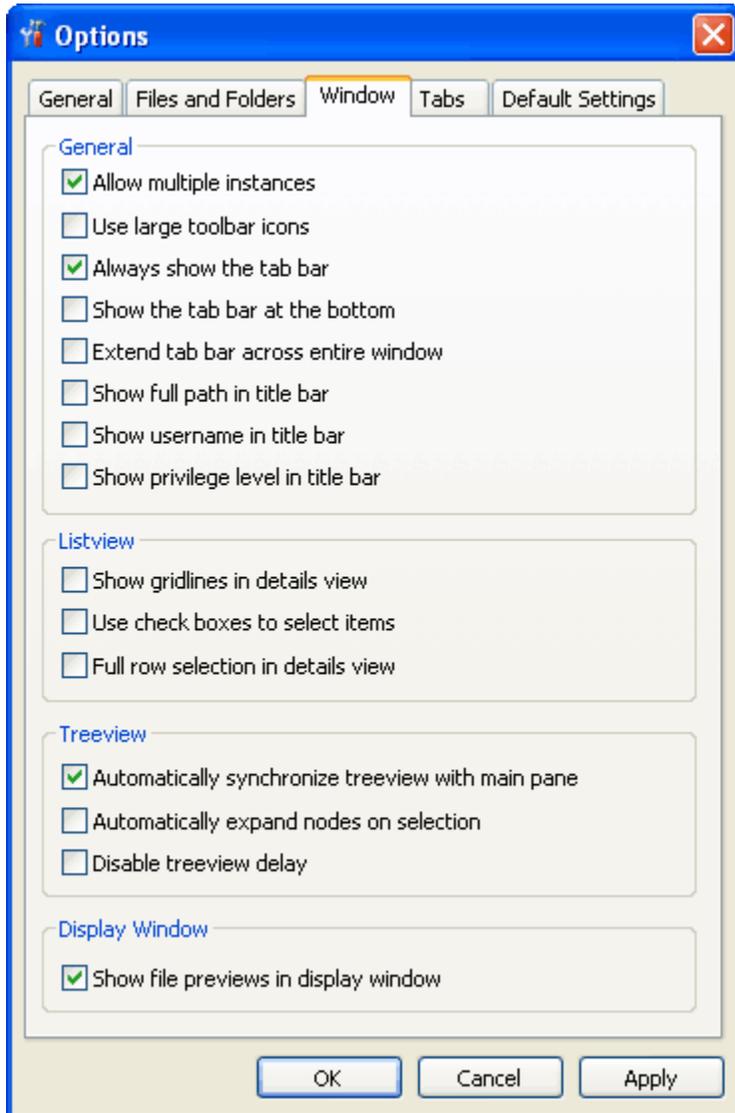
Allow multiple instances

System Default:	Yes

By default, **Explorer++** allows multiple instances of itself, that is, you may open more a one copy of **Explorer++**. Each new copy can have its own set of tabs and settings, etc., subject to information below. If this option is disabled (box unchecked), ie. *do not allow multiple instances*, then attempting to open a second copy of **Explorer++** (eg. by double-clicking on a shortcut) will open a *new tab* instead.

Note: While multiple instances of **Explorer++** are open, each copy maintains tab and settings information in its own memory space. However, on closing, each copy will write complete settings information to the Windows registry (or *config.xml* file). Closing the next copy will **also** do this, overwriting the settings from the first. This means that the

---

last closed instance of **Explorer++** ultimately determines the settings (and tab information) saved for the next session. This behavior is by design.

Tip: If you have multiple copies of **Explorer++** open, the one with your **preferred tabs** (and settings) should be **closed last**. If multiple instances are being used along with *portable* mode, you can make a backup copy of your config.xml file.

Tab bar settings

Always show the tab bar

System Default:	Yes

Disabling this option causes the *Tab bar* to disappear (ie. not displayed) when only 1 tab is open. If 2 or more tabs are open, this setting has no effect.

Show the tab bar at the bottom

System Default:	No

The *Tab bar* is normally displayed at the top of the Files pane; enabling this option (ie. checking the box) displays tabs at the bottom, just above the *Status Bar* (if enabled).

Extend tab bar across entire window

System Default:	No

With this option disabled, the *Tab bar* extends only the width of the Files pane. Enabling this option displays the Tab bar the full width of the window; more tabs can be displayed in this mode.

Title bar settings

These settings modify **Explorer++**'s title bar, which normally only displays the current folder, followed by "Explorer++". A sample title bar follows, showing only the current folder ("AtomExplorer").
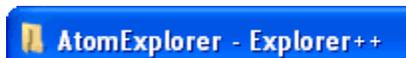
Note that these options may be enabled in any combination. The more options enabled, the longer the window title.

Show full path in title bar

System Default:	No

Enabling this setting displays the full path to the current folder. The following image demonstrates this.

Show username in title bar

System Default:	No

Enabling this setting displays the current user's name (actually the *owner* of the **Explorer++** process), preceded by the domain name. The domain name refers to the location of the user's account, which could be another computer on a network. In most cases, individual computers will just show "DESKTOP" signifying this is a local account/user.

Show privilege level in title bar

System Default:	No

Enabling this setting displays the user's *privilege level*, as determined by the type of his account. This is usually one of

- **Administrators** - high privilege, unrestricted access to all computer functions
- **Power Users** - extensive rights, but may not be allowed certain administrative tasks (eg. take ownership of files)
- **Users** - fewer rights. This prevents accidental changes to the system.
- **Guests** - fewer rights than Users

Full row selection in details view

System Default:	No

Normal selection of an item in details view just highlights the left-most column (usually Name). In order to select that item, the left-most column must be clicked; clicking on another column has no effect.

However, when this option is enabled, the full row is highlighted; selection may be done on any column.

Disable treeview delay

System Default:	Yes

Explorer++ can, if desired, introduce a 500 millisecond delay between selecting a folder in the Folders pane (ie. treeview control) and opening its contents in the Files pane. This delay is entirely *cosmetic*; it may prove useful for users who navigate the folder tree using the keyboard. With the delay, using the up/down arrow keys would not open the folders contents for each folder as the selection was moved up or down.

By default, this option is enabled, disabling the treeview delay; clearing the checkbox enables the delay.

Tabs
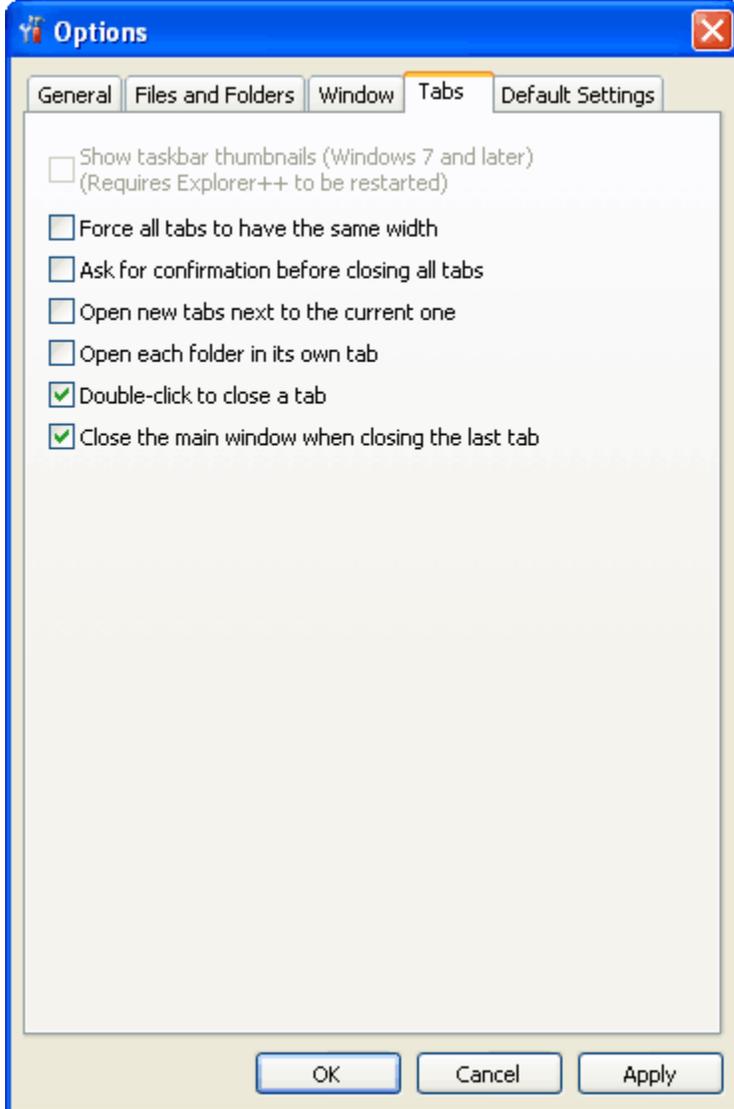
Open new tabs next to the current one

System Default:	No

New tabs on the *Tab bar* are normally created/opened at the extreme right position. This option, when checked, forces new tabs to be created to the **immediate right** of the **active** tab. This position for the tab is used even if the tab is created by a non-active tab elsewhere on the Tab bar using the *Context menu*, for example.

Double-click to close a tab

System Default:	Yes

When this option is enabled (ie. checked), tabs may be closed by double-clicking on the tab in the *Tab bar*. If disabled, the tab must be closed by an *alternate method*, such as the Tab bar's close button.

Close the main window when closing the last tab

System Default:	Yes

Explorer++ usually disallows closure of the last open tab, but with this option enabled, this can be done, and results in **Explorer++** closing completely.

With this setting active, you cannot use the *close* button on the *Tab bar*; you must use double-click (subject to *option*), a key combination (Ctrl+W, Ctrl+F4), or a menu item.

Default Settings
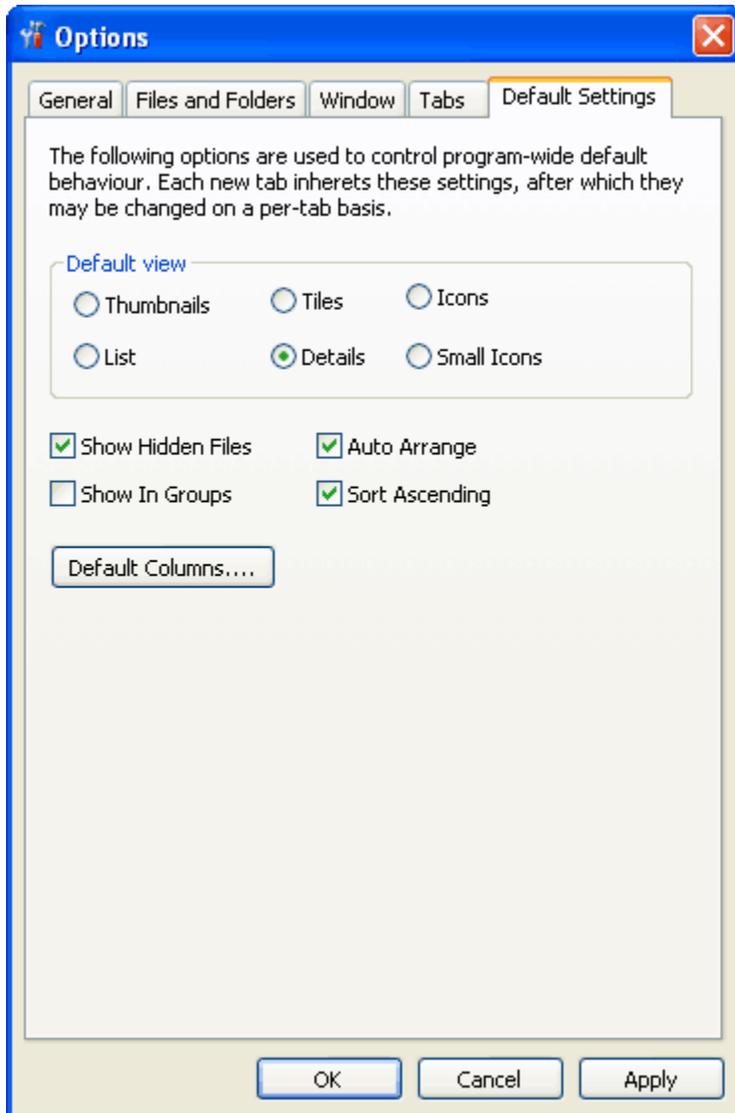
Default Columns. . .

System Default:	see *Folder types*, varies with type

You may set the *columns* (and *column order*) to be shown on **new tabs**; default column widths are always reset to 150 pixels. Default column widths may be changed, however, by the *Save Column Layout as Default* menu item.

About Default Columns and Column Widths

Note: **Explorer++** keeps 2 different sets of *default* columns for use, each with its own column order, visibility and column widths for the 7 different *types of folders*.

- one set of **system defaults**. This set is the set used to create *new tabs*. It can be altered by the **Default Columns. . .** setting (this setting - column order and visibility only) and the *Save Column Layout as Default* setting (order, visibility and width). Column widths are **only** changed when the current tab layout is saved as default. New tabs, when created, adopt this set of columns and widths. These settings are *persistent* , maintained in either the *config.xml file* or the Windows registry.

- one set for **each tab**. This set is used when the tab is switched to one of the 7 different types of folders, but is never used for another tab. Column display settings and width changes are only maintained for this tab. If the **Save Column Layout as Default** option is used, the current column settings and widths (this type of folder only) is saved to the system defaults. Once the tab is closed, these settings are lost.
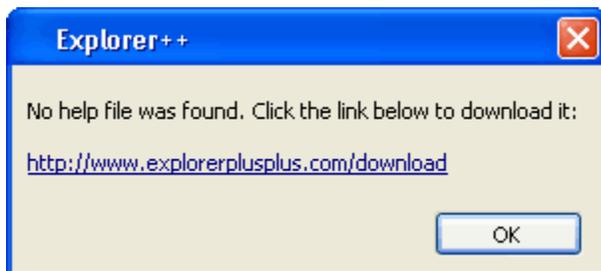

2.9 Help

2.9.1 Help

```
F1
```

This menu item opens the help file at its default page.

You might see the following dialog instead:

which means that you haven't yet *installed* the help file. Click on the link in the dialog to open your browser at the **Explorer++** download page, and download the file (Explorer++ Help.chm) using the **Help File** link. It may be in an archived (i.e. zip file) format; if that is the case, extract it first, then follow the next steps:

- the help file **must** be named **Explorer++ Help.chm** (its default name), and

- the help file must be located in the installation directory, i.e. the directory housing the executable (**Explorer++.exe**). Move it to that location.

- You may need to "*Unblock*" (i.e.. right-click file → Properties → Unblock) the help file after extracting. The file may be blocked from usage by Windows' security routines. It is a **safe** file.

As of the current version of **Explorer++**, (context sensitive) help using the **F1** hotkey is not implemented. Only the Help menu item is functional.

2.9.2 About Explorer++

This simply shows the about dialog.

Toolbars

Explorer++ has 5 internal toolbars, any of which may be enabled (made visible) by checking one of the *menu* items. They are

Tool-bar	ID*	Cus-tomiz-able	Comments
Address Bar	1	no	set or displays the address of the currently active tab (folder)
Main	0	*yes*	provides 19 (in the current version) common tools or actions which may be quickly accessed by clicking a button's icon
Book-marks	2	limited	displays selected bookmarks as buttons/icons along with their name
Drives	3	no	shows and accesses all available volumes (ie. drives)
Appli-cation	4	yes	houses *shortcuts* to selected applications

* IDs are maintained internal to **Explorer++**, but may be useful when editing the Windows registry, or the "config.xml" (ie. *portable* settings) file.

3.1 Moving toolbars

When *unlocked*, he toolbars may be relocated (ie. *docked*) and *resized*, and placed on separate *rows* or combined on a *row* to save space.

Using a toolbar's *grip* (found at the left side of the toolbar), it may be dragged-and-dropped to a new row, or moved along that row effectively resizing the toolbar to its left. Moving a toolbar along a row is only possible when there is more than one toolbar per row.
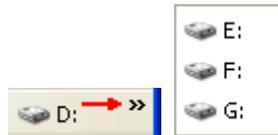
Dragging a toolbar **up** will place it on a row with another toolbar, while dragging a toolbar **down** (to the bottom-most row) places it on its own row.

Toolbar positions are preserved by **Explorer++** for the next session.

Bug

If the **Address Bar** has a second toolbar on the same line to the right of it, it may be partially obscured, even after restarting **Explorer++**.
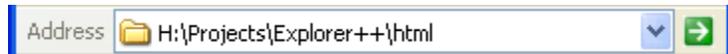
3.2 *Chevrons*

If there is insufficient room to display all toolbar items, a *chevron* symbol (») is used; clicking on the *chevron* opens the rest of the toolbar, as a menu.

3.2.1 Address Bar

`Alt + D`

`Ctrl + L`

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.

The **Address Bar** serves 3 general purposes:

1. it **displays the icon** associated with the current folder (tab),

2. it **displays the address** of the current folder (tab); this address can be copied to the clipboard (if desired), and

3. it allows the user to **enter an address** and, using the *Go* button, **jump to that address**.

Folder icon

This area displays any folder icon associated with a particular folder; the default is usually as shown in the image. Most special folders on the *Go menu* have distinct icons associated with them and will display those icons here.

Tip: Dragging and dropping the **Address Bar**'s icon to a folder (in the Folders or Files pane) - or to another tab - creates a Windows shortcut (ie. a *LNK* file) in that folder. The shortcut points to the original location.

Address/text field

Address display

If a location for the active tab is selected, by any *valid means*, the address/text field of the **Address Bar** displays the current address. If the address is too long to fit in the space allotted (ie. the **Explorer++** window has been resized, and the **Address Bar** is now too small to display the full address, or the address is *deeply nested* in subfolders), the text in the address/text field can be scrolled by placing the cursor in the field, then moving the curser (left/right arrow, Ctrl+left/right arrow, etc.).

Text in the address/text field may be selected and manipulated (eg. copied to the clipboard), by using Windows hotkeys (eg. Ctrl+C for copy), or through the use of a small context (ie. right-click) menu. The *Copy Folder Path* operation copies the entire address to the clipboard.

Address entry

You may enter an address in the address/text field and jump to that address by pressing Enter or clicking the *Go button*. The address must one of either

- a fully qualified path, ie. <drive letter>:\<folder>\<folder>... etc. An example might be:

  C:\WINDOWS\system32\drivers

  Note that fully qualified paths should not end in the backslash (\) character.

  Tip: Every time you enter backslash (\) character, **Explorer++** opens a special (resizable) window and displays a list of the contents (files and folders both) of the path as typed so far (if valid). You may select a folder from this list; **Explorer++** will jump to that folder. Selecting a file will attempt to open it in any *registered* application, the same as if you tried to open the file using a double-click (or single-click, with an *option* set). A *refresh* will restore the true address.

- the name of a subfolder in the current location. The folder must already exist. This is the only time **Explorer++** will accept less than a fully qualified path, except for special locations (following).

- the name of a special location. **Explorer++** recognizes the following locations in the **Address Bar** as special:

Special location	Target
Desktop	*<drive>*:\Documents and Settings*<user>*\D esktop
Documents	*<drive>*:\Documents and Settings*<user>*\My Documents
Pictures	*<drive>*:\Documents and Settings*<user>*\My Documents\My Pictures
Music	*<drive>*:\Documents and Settings*<user>*\My Documents\My Music
Videos	*<drive>*:\Documents and Settings*<user>*\My Documents\My Videos

Entering one of the above locations (case-insensitive) in the address/text field and clicking the **Go** button (or pressing Enter) jumps to the target location. **Explorer++** also recognizes several virtual folders, as shown on the *Go menu*, namely

- *My Computer*

- *Recycle Bin*

- *Control Panel*

- *Printers and Faxes* (opens in Windows Explorer)

- *My Network Places*

although, it is likely easier just to use the Go menu for these locations.

Adress drop-down

The *drop-down* list for the **Address Bar** shows all subfolders (folders only - no files) in the current location. You may click on any folder in the list to put its name in the address/text field; pressing Enter or clicking the **Go** button will take you there.

Go button

The **Go** button, when clicked, set the address for the current tab to the address shown in the address/text field of the **Address Bar**. Pressing Enter is the same as clicking the **Go** button when the **Address Bar** has the focus. The **Go** button has a context (ie. right-click) menu identical to the *Toolbars menu*.

Note: The Tab, Shift+Tab, F6 and Shift+F6 keys cycle the focus in **Explorer++** between the **Address Bar**, Folders pane and the Files pane.

3.2.2 Main Toolbar

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.

The **Main** toolbar provides access to some commonly used **Explorer++** functions. The following buttons are available on the **Main** toolbar, using the *Customize* dialog; your toolbar may not contain all buttons. The table below shows the buttons *loosely* organized by function; your toolbar may have buttons in any order.

But-ton	Tooltip	Brief Description
	Back	returns the tab to the previous location in the tab's history, if one exists
		this (separate) button - part of the **Back** button - opens a menu showing a list of history locations, in the *Back direction*. Clicking on an item returns the tab to that location directly.
	Forward	returns the tab to the newest, but previously visited location in the tab's history, if one exists. This location only exists if the **Back** function has been used; ie. the *Forward history* is only created by the Back function.
		this (separate) button - part of the **Forward** button - opens a menu showing a list of history locations, in the *Forward direction*. Clicking on an item returns the tab to that location directly.
	Up	goes to the parent folder of the current location
	Folders (enabled)	clicking this button toggles the Folders pane off (hidden)
	Folders (disabled)	clicking this button toggles the Folders pane on (visible)
	Copy	places selected items (files and folders) on the clipboard
	Cut	*removes* selected files and folders and places them on the clipboard. Removal is done at the time of *pasting*.
	Paste	pastes data or files/folders from the clipboard
	Delete	deletes files/folders to the Recycle bin (except for removable drives)
	Delete permanently	deletes files/folders without sending them to the Recycle bin
	Properties	displays properties of a file or folder. Multiple files/folders (mixed) may also be used.
	Search	finds files/folders by name and/or attributes
	New Folder	creates a new folder in the current folder (tab)
	Copy To	copies selected files and folders to a different location
	Move To	moves selected files and folders to a different location
	Views	cycles through available views for the current tab
		this (separate) button - part of the **Views** button - opens a menu showing available views and allows the user to select a view directly (menu item is checked)
	Show Command Prompt	opens a command prompt (ie. *DOS box*) in the current directory
	Refresh	re-sorts and re-displays the current Files pane, subject to current settings (ie. Sort By, etc.)
	Bookmark the current tab	creates a new bookmark
	Organize Bookmarks	opens the **Organize Bookmarks** dialog
	Create a new tab	creates a new tab in the files pane

3.2.3 Bookmarks Toolbar

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.

The **Bookmarks** toolbar displays bookmarks (previously created) which have been optionally designated to show on the toolbar by checking the *Show on Bookmarks Toolbar* box. Clicking on the button for a bookmark opens that bookmark (ie. location) in the current tab. Hovering over a bookmark button displays the name and location of that bookmark as a *tooltip*.

Bookmark *folders* may also be shown on the **Bookmarks** toolbar; clicking on the button for a folder opens that folder's bookmarks in a menu.

Context Menu (button)

Each button on the **Bookmarks** toolbar presents the following context (ie. right-click) menu:
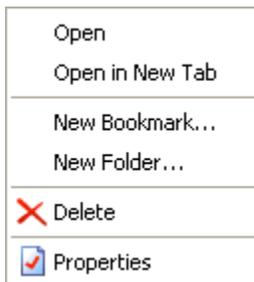
New Bookmark. . .

This function is *similar* to the *Bookmark This Tab. . .* item on the Bookmarks menu, but does not operate on the current tab's location. Therefore, the dialog opens with

- no bookmark name - the user must supply this, and
- no location - the location may be entered manually (if known) or copied/pasted from somewhere else (perhaps from the *Address Bar* on another tab). Any bookmark created with no location will open in the *default new tab folder*.
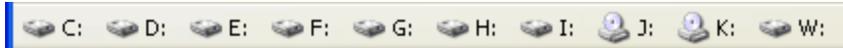
Bug

Bookmarks created by this routine do not show on the *Bookmarks menu* until **Explorer++** is *re-started*.

Tip: Using the middle-mouse button on a bookmark shown on the **Bookmarks** toolbar opens that bookmark in a new tab.

3.2.4 Drives Toolbar

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.

The **Drives** toolbar displays all available *volumes* currently available on your computer. Clicking a button on the toolbar changes the address of the current tab in the Files pane to the root of that drive letter (eg. clicking C: opens C:\ in the current tab). Hovering over a drive button displays the free space and total size of the drive as a *tooltip*, if available (not available if the drive is not ready).

Some users may find it helpful to resize the **Drives** toolbar to its smallest, and dock it on the same row as the *Main* toolbar (right-hand edge). This frees up space, while still providing the functionality of quickly accessing every drive. When the toolbar is docked in this way, only the *chevrons* are visible; clicking on the *chevrons* opens the toolbar as a menu.

Tip: Using the middle-mouse button on a toolbar drive opens that drive in a new tab, however, docking the **Drives** toolbar in this way does not allow this usage on a menu item (see *tip* here).


Context Menu (button)

Each button on the **Drives** toolbar has a context (ie. right-click) menu, with the same functions that would be available from Windows Explorer, or from My Computer (opened in Windows Explorer). In addition, you may open the drive in a new tab from this menu. The functions available on the context menu may vary depending on

- the type of drive (ie. Removable, Local, CD, etc.), and

- other applications you may have installed which affect the context menu.

3.2.5 Application Toolbar

For general information about **Explorer++**'s toolbars, ie. moving, resizing, customizing, etc., see the toolbars *Overview*.
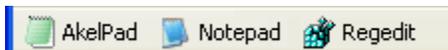
The **Application** toolbar shows a collection of your favorite applications; each button shows the application name and icon, if any. Buttons on the **Application** toolbar are similar to shortcuts to the applications themselves, providing *instant* access to your favorite tools. Unlike true Windows shortcuts, application toolbar buttons may have **no** parameters.

Context Menu (button)

### Open

Clicking this menu item opens the application (corresponding to the button, of course). This function is the same as clicking the button.

Tip: Files from the Files pane may be dragged-and-dro pped onto an application button to open them in that application. Selecting multiple files and using drag-and-drop will open all selected files, providing the application supports this behavior.

Example: multiple text files may be dragged-and-dropped onto the application button for a multi-tab text editor to open all files.

New Application Button...
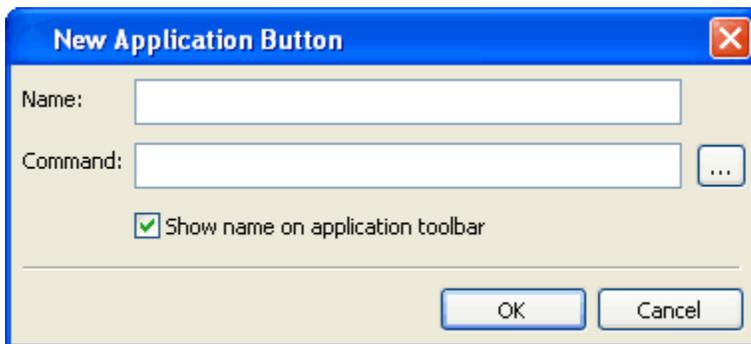
Name

The text entered here is the name for the application that could be shown on the toolbar along with the icon (application specific or Windows default).

Command

This field is the URL to the executable for the application. You may enter the URL manually, if it is known, or *browse* for the executable using the **Open** dialog. If the target file is not an executable (eg. a text or *.txt* file), clicking the application's button on the Application toolbar will open the file in any registered application.

Open (find executable)

This button opens a *standard* Windows dialog (Open) which allows the user to browse for the application executable file. Note that the default file type is EXE, but selecting All Files will browse for other types of files (.com, .txt, etc.). The *Open as read-only* checkbox serves no purpose.

Show name on application toolbar

This box (checked by default) enables/disables the display of the *Name:* field for the target application. Clear this box if the icon is self-explanatory, ie. the nature of the application is clear.

Note: This item (*New Application Button...*) also appears on the **Application** toolbar context menu (ie. right-click over blank space on toolbar).

Delete

This menu item deletes the application button from the toolbar, with a confirmation dialog. It cannot be recovered.

Properties

This function opens the properties for the application *shortcut* and button. A dialog (*Edit Application Button*) is opened which is identical in function to the *New Application Button*, with all pertinent fields (for the button) already filled in. Fields may be edited from this dialog.

Tip: **Re-ordering buttons on the toolbar** (advanced - be careful!)

Application buttons are normally placed on the Application toolbar in the order they are created. Through judicious use of the Regedit application, they may be re-ordered.

1. open Regedit and navigate to `HKEY_CURRENT_USER\Software\Explorer++\ApplicationToolbar`

2. select the **ApplicationToolbar** key in the left panel, and **Export** its contents to a suitable location using the context menu

3. edit the file to renumber the sub-keys (application buttons). They must always be numbered 0, 1, 2, etc. The first button is 0, the second is 1, and so on. The actual order they appear in (inside the file) is not important.

4. **make sure Explorer++ is closed**

5. Double-click on the edited file to re-enter the data into the registry.

6. That's it!

A similar technique applies to using **Explorer++** as a portable application - edit the **config.xml** file to re-order the application button lines. A number is not used; the order of the lines (under <ApplicationToolbar>) determines the button order.


3.2.6 Lock Toolbars

This menu item - whether on the *Main* menu (*View/Toolbars*), or on the context menu of one of the toolbars themselves - toggles the **locked** state of all toolbars.

- **Locked** - toolbars are not movable, and have no grips (see *Moving toolbars* for image of *grip*)
- **Unlocked** - all toolbars have grips, and are movable.

Although it does not affect functionality, it is recommended that toolbars are locked, once positioned, to avoid any accidental movement, etc.

Tip: The *Main toolbar* may be *customized*, even in the locked state.

3.2.7 Customize...
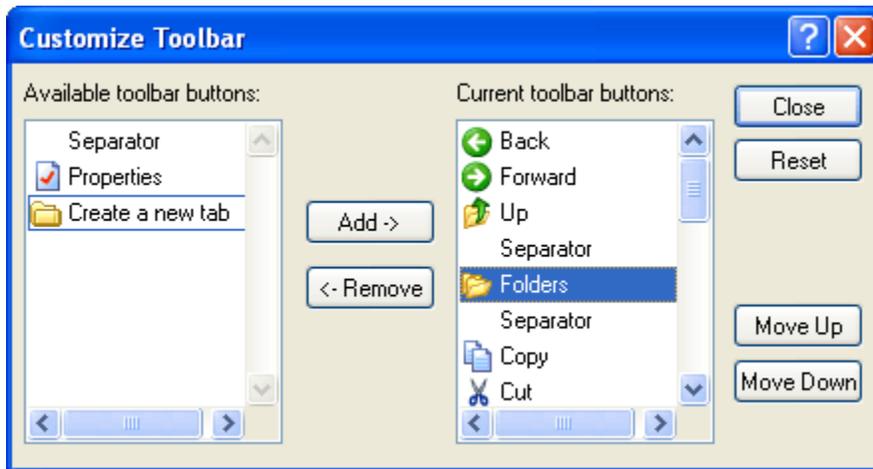
The **Customize Toolbar** dialog manages buttons on the *Main* toolbar by

- adding a button
- removing a button or separator bar
- re-ordering buttons.

Available toolbar buttons

This pane shows the toolbar buttons that are available, ie. have not been included in the current toolbar configuration. The *Separator* item is always available. For more information on available buttons, see the *Main* toolbar.

Add -›

Clicking this button adds the currently selected tool (in the **Available** pane) to the toolbar. The button is inserted above the selected item in the **Current** pane.

‹- Remove

Clicking this button removes the currently selected button (in the **Current** pane). The button removed becomes available in the **Available** pane, but is no longer shown on the toolbar.

Current toolbar buttons

This pane shows the buttons (and *Separators*) currently assigned to the Main toolbar. The button at the top will show as the left-most button on the toolbar.

Close

This button closes the dialog, preserving the current layout of the Main toolbar.

Reset

This button resets the Main toolbar to its default button set.

Move Up

This button moves the selected item (in the **Current** pane) up in the list. This has the effect of moving that item one position to the left on the Main toolbar.

Move Down

This button moves the selected item (in the **Current** pane) down in the list. This has the effect of moving that item one position to the right on the Main toolbar.

Panes

4.1 Tab bar

The **Tab bar** displays tabs which are currently available for use; each tab is essentially a separate Files pane, viewing a specific location.

The user may quickly switch from location to location simply by selecting the appropriate tab.

Each tab has a (right-click) *context menu*.

4.1.1 Viewing tabs

If more tabs are open than will fit on the tab bar (depending on current sizing of **Explorer++**'s window and other panes), a set of scrolling buttons are available, allowing you to scroll left or right to view/select other tabs which are currently not visible.

Tip: The Tab bar may be extended to the full window width using an *option*.

Note: Hovering the mouse over a tab displays it's true location, regardless of the tab's title. Tab's opened at virtual folders show their *CLSID* (class ID), Windows' internal representation of their location. Practically, these *CLSID*s are of no use, except to a programmer.
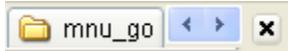

4.1.2 Opening new tabs

A new tab may be opened using a variety of methods, as listed *here*. From the **Tab bar**, the easiest methods may be

- double-clicking empty space on the bar. This opens a file displaying the *default contents*.

- using the tabs *context* menu

4.1.3 Closing tabs

When hovering the mouse over the "X" on the right side of the **Tab bar**, it becomes a "Close" button which will close the current tab. Tabs may also be closed by double-clicking on a tab, subject to an options *setting*. Other methods of closing a tab are:

- from the *File menu*
- from the tab's context menu. The tab need not be the current (ie. selected) tab.

Tip: Be careful closing tabs - the **current** tab may not be visible! A tab closed in error must be re-opened and re-configured, as necessary.


4.1.4 Re-arranging tabs

Tabs may re re-arranged using *drag-and-drop*, that is

- click on a tab and hold the left mouse button down
- move the cursor to the new (desired) position on the Tab bar
- release the left mouse button

4.1.5 Moving files and folders

Files and folders in a tab's file view (ie. the Files pane) may be moved from one folder to another simply by *dragging* and *dropping* them onto a tab. If you pause slightly before *dropping* the items onto the tab, that tab will become active, but this is not necessary in order to complete the move.

4.1.6 Context menu
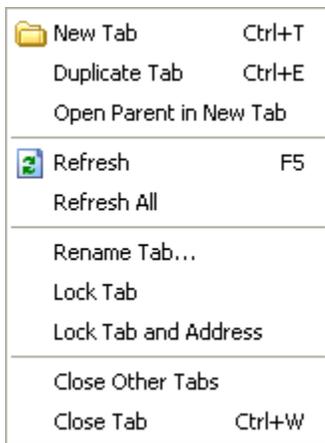
Duplicate Tab

This item opens/creates a *new tab* on the tab bar, but duplicates the tab from which this item was called. Placement of the tab is dependent on an *option* setting.

Note: Duplicating a tab does not duplicate it's location *history*.
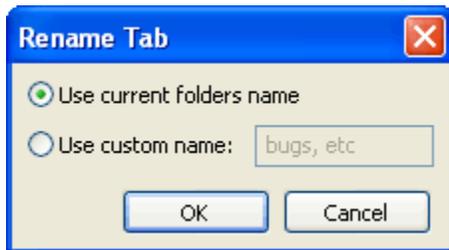

Open Parent in New Tab

This item opens/creates a new tab on the tab bar, using the location of the current folder's immediate parent. This is the folder that *houses* the current folder.

Note: The *Desktop* has no parent; this operation will **not** open a new tab if the current location is Desktop.

Refresh All

This item performs a *refresh* on all tabs currently open.

Rename Tab. . .

Ordinarily, a newly created tab assumes the name of the actual folder it displays (minus the drive and path information). This item, however, allows you to rename the tab to something more suitable to your needs. For example, while working on a programming project, you may have a tab open at "C:\Project\today\revisions\code", but other folders of the name "code" may also exist for that project (eg. different versions, dates). You can rename the folder to "Today's code" by selecting this item, choosing "Use custom name:" and typing in "Today's code". Renamed tabs are *persistent*, that is, they will remain renamed even during other sessions.

Note: On first opening the *Rename Tab* dialog, the "Use custom name:" setting is always selected as a convenience; it is assumed that you want to set a custom name.

Lock Tab
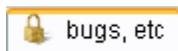
This item locks the tab, preventing it from being closed. The tab's icon is changed, and the menu item becomes checked, reminding the user that the tab is locked. Locked tabs can still be *re-arranged* or *renamed*. Locked tabs must be manually unlocked to restore default behavior. Tab locked status is preserved between sessions.

Lock Tab and Address

This item locks the tab, as above (ie. can't be closed), but also prevents the address from being changed by double-clicking on a folder or drive. A new tab is opened instead, displaying the desired new address.

Close Other Tabs

This item closes all tabs on the Tab bar **except** the tab whose context menu you are using, even if that tab is not active. Of course, when all other tabs are closed, the remaining tab becomes active.