

# xplorer<sup>2</sup> user manual

**xplorer<sup>2</sup>**

Exponential growth to file management productivity

**ZABKAT**

## Introduction

Whether you work with documents in your office or school, or collect photos as an amateur, you will have lots of files. Windows explorer is a basic file manager that will soon annoy you with its inconsistencies and limitations. If you deal with many files and folders you need a heavy duty file manager like xplorer<sup>2</sup>, which will help you be productive, with almost every kind of file you possess:

- Find your files wherever they may be (hard disk, network, cloud storage or mobile phone)
- Work in large folders with many files, using focused filtering and file selection methods
- Quickly preview most file type contents (text, image, audio/video)
- Browse many folders simultaneously with dual panes, folder tabs and Miller columns
- Manage multiple files in one go (copy, delete, rename, change dates and attributes)
- Check for file modifications, synchronize folders and perform safety backups
- Maintain a tidy organized file system clean of duplicate files and wasted space

If you have worked with windows explorer, then you will feel immediately at home with xplorer<sup>2</sup> as it supports the same tree/view folder browsing, view modes like details and thumbnails, mouse drag drop and context menu operations. The more advanced xplorer<sup>2</sup> features will require some learning curve, but the interface is simple and consistent, and there's plenty of help and support, both inside the program and online.

xplorer<sup>2</sup> integrates fully with Windows Shell model, so it can browse in all locations, even within "virtual" folders like compressed archives and SFTP. It can utilize more than 300 system file properties and integrates with any [shell extensions](#) you install to see more properties, icon overlays, document previews and so on.

With its small size (around 3 MB) and low system resource use, advanced multi-threaded multi-window design for speed and efficiency, customization, a clean minimalist interface and keyboard shortcuts for most frequently used commands, we trust you won't be disappointed for choosing xplorer<sup>2</sup> for your file management needs. In fact chances are you'll never look back to explorer or any other file manager!

## How to use this guide

xplorer<sup>2</sup> does many things for you so unavoidably its operation manual is huge. Understandably you don't want to read the entire help file from start to end — it is recommended to read the [overview](#) chapter at minimum though. See the table of contents below for the basic organization in chapters and sections. Use the **search** function of your browser (usually pressing <CTRL+F> keys) to search for a keyword — ideally search for the exact menu command name you are interested in — or see the [menu index](#) in the appendix.

This HTML document has many internal cross references and links that jump to other parts of information, possibly far away from what you were reading. To come back where you were (before clicking on any [anchor](#)), use your browser's **Back** button or press <Alt+LeftArrow> keys.

At major section boundaries, you will find links to jump to the previous and next sections, and the table of [Contents](#).

Each section begins with a quick overview of the subject at hand, followed by more in-depth information. Various popouts are sprinkled all over the place with extra information or warnings. Most sections include further reading articles and demo videos, which you recognize by these symbols:



Extra information and tips for increased productivity



Warnings whenever an operation is potentially dangerous or confusing




Advanced information for expert users, which may be ignored by most



Links to demo videos, which may be more to your liking if you prefer video explanations to reading manuals

Instructions often refer to commands in the menu bar and are shown in **bold print**, including the menu "full path", the names of all main and submenus you need to open to access the command. Menu paths are separated by > symbols. E.g. **File > New tab** means you open **File** from the main menu and find **New tab** somewhere in it.

Some explanations include keyboard shortcuts you press to invoke a command, as an alternative to using the menu. Keyboard shortcuts are placed in angle brackets and can combine more than one keys, e.g. <CTRL+C> means you must press CTRL and C keys together.

Most of the information herein regards the professional xplorer<sup>2</sup> edition. A few advanced commands are only available in [ultimate](#) edition and are clearly marked as such with  icon. Note that many features are missing from the free lite version. Such commands are marked with **[P]** in the lite version menu system



Font too small to read? Press CTRL+NUMPLUS keys to make it bigger

# TABLE OF CONTENTS

**Program interface**  
[Working with files](#) | [Dual panes](#) | [Moveable panes](#) | [Folder tree](#) | [Status information](#) | [Ribbon toolbar](#) | [Workflow](#) | [How to get help](#)

**Browsing the shell namespace**  
[Folder types](#) | [Folder bookmarks](#) | [Column mode](#) | [Namespace extensions](#) | [Deep path support](#) | [File properties](#) | [Folder views](#) | [Group mode](#) | [Folder view settings](#)

**Simple file management**  
[Rename](#) | [Recycle and delete](#) | [Advanced drag-drop](#) | [Previewing and launching items](#) | [Extract information](#) | [Changing properties](#)

**Industrial folder management**  
[Filtering visibility](#) | [Selection engine](#) | [Defining hyper-filters](#) | [Quick filter/selection box](#) | [Manage color tags](#) | [Color coding filetypes](#)

**Scrap containers**  
[Populating scrap panes](#) | [Managing collections](#) | [Mini scrap](#)

**Advanced file management**  
[Advanced copy/paste](#) | [Robust file transfer](#) | [Robust deletion](#) | [Mass renaming](#) | [Changing timestamps and attributes](#) | [Splitting and merging](#) | [File and folder comments \(tags\)](#)

**Examine and maintain your filesystem**  
[Comparing & synchronizing folders](#) | [Checking file transformations](#) | [Visualize disk space usage](#) | [Detecting duplicate files](#) | [Photo cleanup](#) | [Fuzzy groups](#) | [Convert groups to folders](#)

**Search for files and folders**  
[Windows Desktop Search](#)

**Automation of DOS (and windows) commands**  
[\\$-Tokens](#) | [Command output redirection](#) | [Automatic script generation](#) | [User commands](#) | [Optional command line arguments](#)

**User interface automation**  
[Macro commands](#) | [Programmable column](#)

**User Interface customization**  
[Keyboard customization](#) | [Toolbar customization](#) | [xplorer<sup>2</sup> context menu](#) | [Program options](#) | [Window layouts](#)

**Installation and registration**  
[Ultimate edition](#) | [Program registration](#) | [Suspected bugs](#) | [Troubleshooting](#)

**Appendix**  
[Keyboard shortcuts](#) | [Stock properties](#) | [editor<sup>2</sup>](#) | [All menu commands](#)

Table References

[1a.Special Character](#) | [2.Compare Links](#) | [3.Reusable Command Tokens](#) | [4.External Commands](#) | [5.xplorer<sup>2</sup> Cmdline Args](#) | [6.Macro References](#) | [7.Helper Function](#) | [8.Compare xplorer<sup>2</sup> editions](#)|[9.Problem Solutions](#) | [10.Keyboard Shortcuts](#) | [11.File Properties](#) | [12.Main Window Menu](#) | [13.Scrap Windows Menu Commands](#)

## Program Interface overview ▶ PLAY

xplorer<sup>2</sup> window presents the contents of one or more folders, and offers a palette of file management commands to work with the files you see. The figure below demonstrates all the window features — it may look complicated but you can turn off GUI elements you don't need using **View** menu.

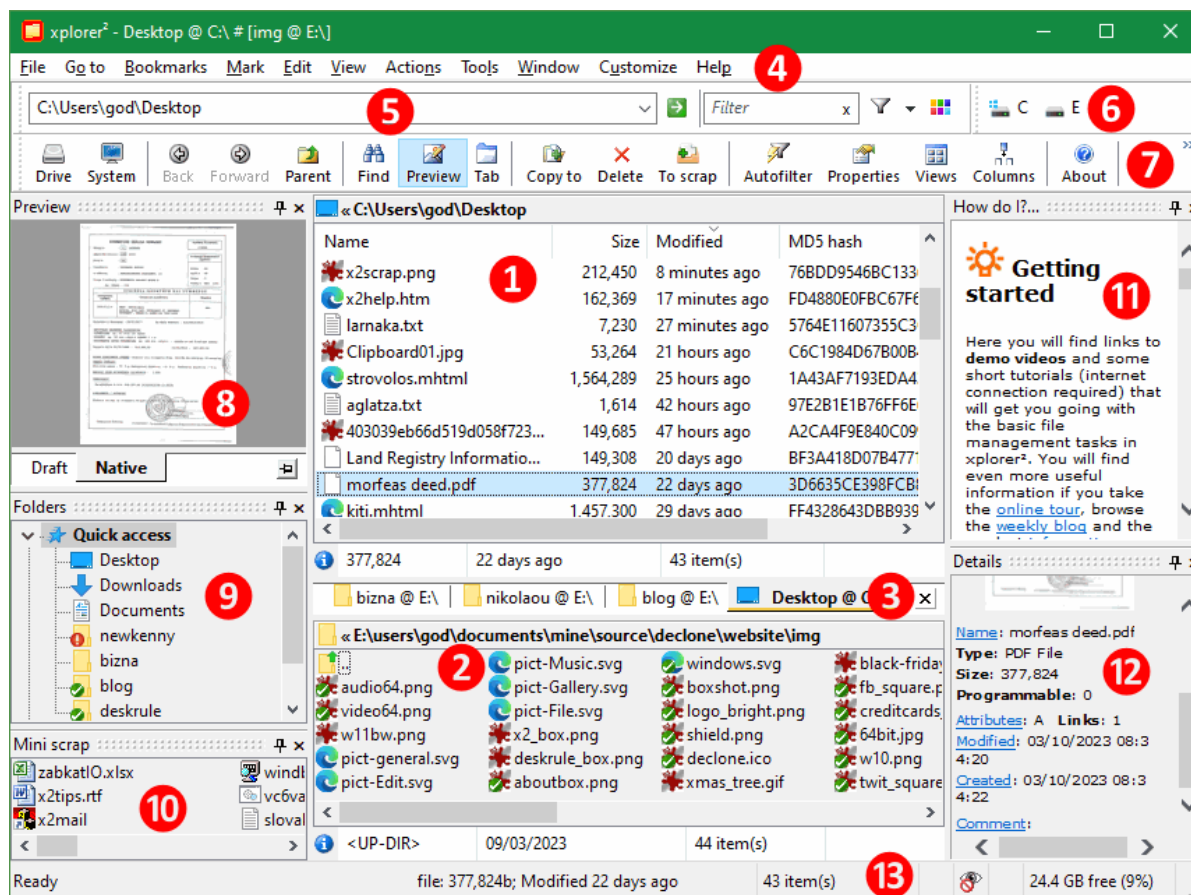


Figure 1. The main window


Here is a brief description of the main window elements, by their reference number in the red circle:

- ① **Active pane.** Contains the items in the active folder, the one you are working with. Each file is displayed in a list with its name and an icon representing its [type](#), as in windows explorer. In this example the view is set to details, showing file information in columns. The pane's titlebar is showing the folder's full path, whose parts are [clickable](#). The active pane is drawn with a brighter background color for easy identification. All menu and toolbar commands operate on items in the active pane.
  - ② **Second pane.** In dual pane mode, you get to browse another folder, facilitating easy file transfers. The inactive folder behaves like the active pane, only it shows with a darker background color. The inactive pane will become active if you click on it or press <TAB> key, deactivating the previously active pane. The two folder panes are independent and can be setup differently (e.g. in the figure the second pane is set in list view mode). If you don't need the second pane, turn it off using **View > Dual pane** menu command.
- The menu bar is organized in categories. There are top level menus like File and View, which open up submenus, and some items in those may open a sub-sub menu, and so on. The separator > is used to refer to menu command "paths" to guide you finding them. E.g. **View > Dual pane** means you open the View menu and use Dual pane command inside it.
- ③ **Folder tabs.** ▶ PLAY Each folder pane can host multiple folders in tabs (there are 4 shown in the figure). You open a new tab using **File > New tab** menu command. When first created, a tab inherits the active pane's properties (folder browsed and pane style), but from then on it is independent. You can change the style to something else and browse to different folders. The label of each tab shows the folder name (the full path is shown as a tooltip). Tabs let you switch between folders quickly, but note that only one can be **active** at a time. Click on a different tab to activate its folder. Use **[X]** button to the right of the tab strip to close the active tab.
  - ④ **Menu bar.** The standard windows menu bar holds all the program commands, organized in standard file management categories. Each menu item shows a brief help message on the statusbar ⑬ explaining its use. Popular commands show a little icon, and if there's any keyboard shortcut (accelerator key) it is also shown. If you don't use most of the commands, you can switch to **Plain menus** ticking the relevant [program option](#), then only the most basic commands will show.
  - ⑤ **Address bar.** Here you type the path (address) of the folder you want to browse in the active pane. It holds a history of recently and frequently browsed folders in a drop-down section (use the down-arrow to access them). xplorer<sup>2</sup> addressbar isn't clickable but you can use each folder's titlebar panel for the equivalent [functionality](#). Addressbar also accepts typed DOS and windows [external commands](#) to execute. Here you will also find the [quick filter](#) box where you can quickly filter, select and search for files in the active folder. Right click on any toolbar to turn the addressbar on/off using the context menu.
  - ⑥ **Drivebar.** This toolbar shows all top level disk drive letters, both internal and externally connected via USB cable, as well as mapped network drives. Click on any drive button to browse the root of said drive. Root drives are also available from a pane's [titlebar](#). Use **View > Toolbars > Drivebar** submenu to turn the drivebar on/off.
  - ⑦ **Command toolbar.** Holds the most frequently used menu commands as buttons for quick access. If you right click on the toolbar you can use the context menu to turn buttons' **Text labels** on/off, use large or **Small icons** and [Customize](#) which buttons appear on the toolbar (any menu command with an icon can be easily placed on the toolbar).
  - ⑧ **Quick Preview.** This resizable and [detachable](#) panel shows a preview of the selected file, if possible, depending on the content. So for photos you will see a thumbnail, for videos a mini media player and so on. The draft [preview tab](#) is for quick and dirty previewing; switch to **native** preview tab for more accurate office and PDF previews (including formatting, not just plain text). Right click for the context menu which allows to search for text, choose the preview font and some other text


content related menu commands. Use **View > Quick viewer** menu command to turn this pane on/off.  
TIP: Press both mouse buttons on a file to see its draft preview in a big, easy to dismiss [peek preview](#) popup window

- **⑨ Folder tree.** Shows a hierarchical tree representation of your entire filesystem, including local and network folders. Use the arrows to extend (or collapse) tree branches and reveal subfolders. Optionally you can manage a [quick access list](#) of your mostly used folders for easy access in a separate tree branch. A number of [tree options](#) exist, e.g. showing two trees in dual pane mode, automatically aligning the tree with the active folder and so on. Use **View > Show tree** menu command to show and hide the tree.
- **⑩ Mini scrap.** Here you can drag-drop files and folders for easy access, e.g. files or programs you frequently open, folder bookmarks etc. [Scrap](#) windows are generic containers for files and folders that can be located anywhere in your filesystem; the miniscrap is a full file manager pane but it is mostly used as a quick access list. Activate using **View > Mini scrap** menu command.
- **⑪ Help videos.** You will find a selection of demo videos that explain how the program works. It is recommended you watched at least the first few ones that explain the basic window and file operations. **Help > How do I?** menu command turns this pane on/off.
- **⑫ Details pane.** Shows a small thumbnail preview and type-specific information for the selected file in the active pane. For example, for images you see their dimensions, for movies the total duration etc. Some basic properties can be changed by clicking on the underlined "links", offering shortcuts to the equivalent menu commands. This pane can be [customized](#) to add any file property you want to see. Enabled using **View > Details pane** menu command.
- **⑬ Status bar.** The bottom edge of the window shows information about the active item and any selected files and other [status information](#) as the presence of a visual filter that hides some of the folder contents. Hover the mouse pointer over a status bar pane to see what it does in a tooltip. When you issue file management commands, the progress and results (or errors) are shown on the statusbar as a message that automatically goes away after a while. If you don't want to see the statusbar, turn it off from **View > Toolbars > Status bar** menu; in such a case error messages will show up as messageboxes.

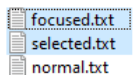
You only work with one of all these panes at a time. The easiest way to switch activation is to click on a pane with your mouse, but there are also [keyboard shortcuts](#) that jump to particular panes — e.g. <SHIFT+TAB> focuses the addressbar **⑤**.


As with all windows programs, xplorer<sup>2</sup> can be resized or maximized to see more file contents. Within the window, the dual panes and all the peripheral panes like the tree can be resized using **splitter** bars. Place your mouse over the lines dividing the panes, and it should turn into a resizing cursor ; then you can grab the splitter with your mouse and move it left/right (or up/down) to change the relative pane sizes.

As in windows explorer, with xplorer<sup>2</sup> you work with one folder at a time, which is the active tab of the active pane (either **①** or **②**). The active folder path is shown in the addressbar **⑤** and in the active pane's titlebar (the window titlebar is showing summary paths of both active and inactive panes). Use the folder tree, drivebar or type paths in the addressbar to browse a different folder.

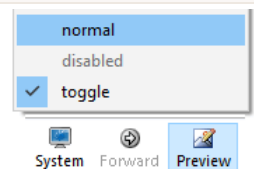
 xplorer<sup>2</sup> dual panes share the auxiliary panels like drivebar and tree. You won't see two separate drivebars for example; whenever you click on a drive letter it will affect the active pane; hold down <ALT> key as you click on a drive button to change the folder in the *inactive* pane. <ALT> modifier works for all browsing actions.

In the active folder pane you **select** one or more files with the mouse or the keyboard (or using a command from [Mark](#) menu) to operate on them, using a menu or toolbar command. Selected items show with a blue background color. In multi-selections, a single item is the **focus**, which is drawn with a dotted rectangle as in the pic to the right. It is usually the item you clicked on last. The focused item is special as it is the one that gets previewed in panel **⑧**, whose details show in pane **⑫** and the statusbar. When only one item is selected, it is also the focus obviously.




 Selecting multiple files can be tricky, as one accidental click may unselect all files. xplorer<sup>2</sup> doesn't offer **checkboxes** for selection; for safe selection use **Mark > Sticky selection** menu command. When this mode is active, clicking on an item toggles its selection state without affecting any previously selected items. Use **Mark > Sticky selection** again to turn this mode off and resume normal item selection.

Many commands require a selection to work. If there's nothing selected then the command is **disabled**; the menu command or/and toolbar button appear ghosted to indicate the disabled state. Other commands are toggles, turning a feature on/off: when enabled a tick appears on the menu item — or the toolbar button appears depressed. On the right you can see examples of normal, disabled and ticked menu and toolbar states



Command availability also depends on the [folder type](#) being browsed; some commands are only available in regular (filesystem) folders and will be disabled if you are working in a special folder like ThisPC or Control Panel.

xplorer<sup>2</sup> uses **context** sensitive menus depending on where you right click on. It presents the commands that are relevant to the area you clicked. For example if you right click on a tab you see tab-related commands, if you right click on the list header you see commands related to columns and so on. Thus you don't need to search among the hundreds of main menu bar **④** available commands; always try a right click first.

 xplorer<sup>2</sup> is a DPI aware program, so it should pick a good size font for all the user interface elements depending on the text size you have for your system display settings. When you move an xplorer<sup>2</sup> window to a different monitor, it will also automatically resize itself to match the DPI settings in that monitor. If you nevertheless want to use a bigger or smaller font, you can do so using the [main font](#) program option.

xplorer<sup>2</sup> user interface is using standard windows OS components that are easy to automate with accessibility tools (NVDA/JAWS for visually impaired users).

#### Further reading

- xplorer<sup>2</sup> main window flexibility: screenshots with possible [pane arrangements](#)

## Working with files

A standard list view windows control shows the files (and subfolders) of the active folder. This is exactly as the content pane you see in windows explorer, and behaves the same, e.g. a double click on an item will open it. Here are the main parts of the active pane:

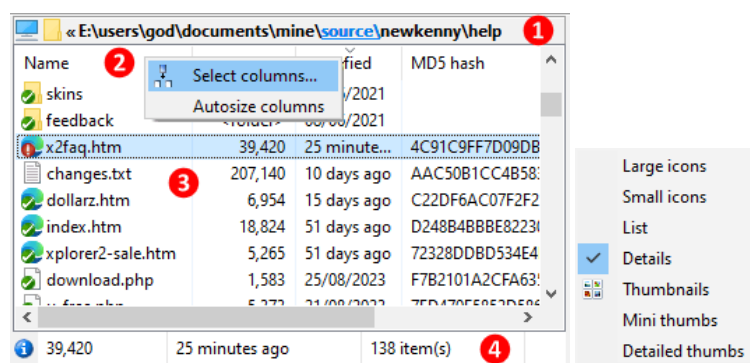


Figure 2. The active folder pane (and view mode submenu)

- ① **Titlebar.** Shows the full path of the browsed folder. The path segments are clickable, offering an easy way to browse a parent folder. If you **right** click on a path segment it pops a menu with its subfolders allowing you to select any neighboring folder. The leftmost ThisPC icon pops a drive selection menu for all the local and external drive letters when clicked.
- ② **Column header.** When you see file details like name, size and date modified, the list header shows the column names and [sort order](#) information. Click on any column to change the sort method and ascending/descending sort direction. In the pic above pane contents are sorted by date modified (descending), as you can tell by the little arrow over the column **Modified**. Drag the column header splitters to resize a column, or double click on the divider to automatically size the column based on its file contents. If you right click on the header you can **Select columns** from the context menu. There are hundreds of [file properties](#) to choose from.
- ③ **Folder contents list.** Shows the files and subfolders of the browsed folder, see below for more details.
- ④ **Infobar.** This is like a separate [status bar](#) for each pane, showing properties for the focused (selected) item. Right click on the infobar to choose the columns you want to see. It also can show information about the selected items. Use **View > Toolbars > Info bars** menu command to turn it on/off.

The folder contents are shown with file type icon and the filename, and possibly more details. Use **View > Pane style** submenu to choose the preferred viewing mode (this is also available from the background context menu, when you right click on a blank spot). The most important view modes are:

- Details** (as in the snapshot above). Shows file properties, each file takes a whole line
- Thumbnails.** This mode is for folders with lots of photos or videos, where you get to see a thumbnail preview and a filename only. You can combine details with a small thumbnail using **Detailed thumbs** view mode.



Thumbnail size is [customizable](#). There is an additional **Mini thumbs** mode that shows more (but smaller) preview icons. You can dynamically **zoom** thumbnail sizes using <Ctrl> key with your mouse wheel. Thumbnail spacing is controlled from [advanced options](#).

- List mode.** This only shows a small icon and the filename, but it is the most compact, allowing you to see many items simultaneously. It can be combined with infobars or the [details](#) pane to see properties for one item (the focus).

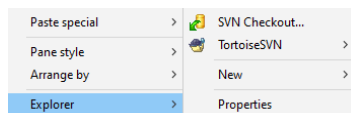


The column size in list mode may be as large as the *longest* filename that happens to reside in the folder. To avoid oversized columns, use *"Max list column width"* [option](#) to provide a sensible column width limit. If any long file name is cropped, you can see it full by hovering the mouse pointer over it (via the infotip)

If a folder contains too many items, horizontal or vertical **scroll bars** appear so you can see all the contents. You can quickly jump to a particular file name if you start typing a few *starting* letters of the name rather quickly. If you type 4 or more letters then you activate the [quick filter](#) box that will show you only files that match the name fragment you typed.

You manage files by clicking on them to select them, then use any toolbar or menu command. Right click on the selected files to show the standard **shell context menu** with windows explorer commands like copy, delete, rename etc. You can **drag** selected items with the mouse and drop them to other locations e.g. move them to different folders.

Another standard shell item is the **background** menu. Find an empty spot (without any text information) and right click to see the menu. If you have difficulty showing the background menu — can't find where to click? — then right click on the folder icon on the pane's titlebar ①. In there you will find **Explorer** submenu with more shell related commands, e.g. create new word documents etc. To see extended commands press <SHIFT> key as you right click.



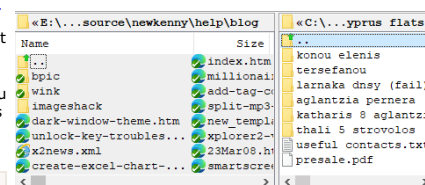
If you want to manage files from many folders simultaneously, use a [scrap window](#). You can hand-pick items to add with mouse drag-drop or extract entire folder hierarchies with **File > Browse flat** menu command, which will expand a folder including all its subfolders, all the way down.

## Dual panes and tabs

When transferring files you need to see both source and destination folders. That's why xplorer<sup>2</sup> shows two folders simultaneously, so you can copy files easily. Dual panes are also useful for [comparing](#) and synchronizing folders. xplorer<sup>2</sup> also allows browsing many folders simultaneously in tabs, but only two folders will be visible at any given time. Inactive tabs are hidden from view.

When you are not copying or comparing folders, you can turn off the dual pane mode with **View > Dual pane** menu command, giving you more room to work with folder contents. Use the same menu to turn dual panes back on. The [main window](#) figure above shows the dual panes stacked vertically, but you can also place them side by side with **View > Tile horizontally** menu. Drag the splitter (divider line) to make one pane bigger than the other; <CTRL+E> keyboard shortcut will make the panes equal size.

Only one pane is **active** at a time, the one with lighter background color (the left one in the picture to the right). All menu commands operate on the active pane. The right hand side pane is inactive, drawn with a darker background color. Press <TAB> key to make the other pane active — or click on it with the mouse. The background colors will swap, signalling the change of activation.



If you have trouble recognizing which pane is active, you can choose a darker color for the *"inactive background"* through [settings](#). For extra visibility tick *"Highlight active pane's titlebar..."* option.

Usually both panes are set in the same style, e.g. both showing details, however you can configure the 2 panes independently. Click on the first pane to make it active and choose a mode from **View > Pane style** menu, e.g. **List**. Then click on the other pane to activate it, and choose its view mode from the same menu. Remember all file management occurs within the active pane, whichever it may be. It is the single **focus** of your workflow.

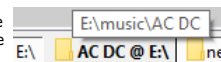
You can add more folders using **File > New tab** menu command. A freshly opened tab inherits the folder and view properties of the previously active tab, but from then on it is totally independent. You can browse different folders, change its style to thumbnails, or whatever you desire.





You can create new tabs from the folder tree and the miniscrap pane too, if they happen to be active. Whichever folder you have selected in the tree will be opened in a new tab. If you have more than one folder selected (this is possible in the miniscrap), then **multiple** tabs will open, one for each item selected.

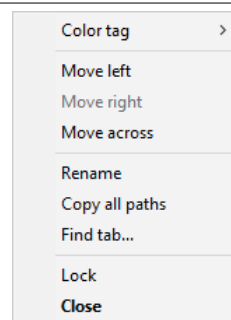
You probably have worked with tabs before (e.g. your internet browser) so you will know about the little tabs that represent each folder. Only one tab is active at a time; click on a different tab to bring up another folder (and simultaneously hide the previously active folder). You can add tabs for both the left and right (dual) panes, however only the 2 visible folder tabs are in business (one active, one inactive). You can open as many tabs as you like, but if you have too many of them, it may get confusing — use **Window > Find tab** menu command in such cases to find a particular folder by its (partial) name.

Each tab shows the folder icon and its name. To see the full path, hover the mouse on a tab and a popup window will appear showing the path. The **active** tab is drawn with an orange highlight to tell it apart (if you cannot see this color use [advanced options](#), *"Folder tab highlight"*). You can have [tabs on top](#) or to the bottom of the pane to suit your style.





To close a tab you can double-click on it or press <CTRL+F4> keys. Use **Window > Close all tabs** menu command to close all tabs except for the active one. If you close a tab by accident, you can bring it back using **Window > Reopen tab**. You will find more tab related commands if you right click on a tab to see its context menu.


- You can **move** a tab left or right using the respective menu commands. You can also drag-drop a tab to a different location. **Move across** command sends the tab to the inactive (dual) pane.
- You can make a tab **stand out** by giving it a color tag (this actually tags the [folder](#)) or **Rename** it — then you see your own tab title instead of the folder name.
- **Copy all paths** menu command copies all the paths as clipboard text. Paths are separated with commas. One possible use is to start a [file search](#) inside all tabs.
- When you **Lock** a tab, then you freeze its location. Locked tabs will not change folders. If you attempt to change folder, a new tab will open. Locked tabs show a little lock  instead of the folder icon. Use Lock menu again to unlock a tab. For more information here's a [demo video](#) .



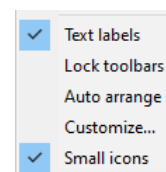
You can **save** a set of folder tabs using **Customize > Folder groups > Add current** menu command. Then you can easily restore the set, picking it from folder groups menu again. The same menu appears when you right click on the empty space past the last tab (or on the **[X]** button to the far right of the tab strip). The list of tab sets is managed in the same way as [bookmarks](#), so you can make them easily accessible with **Organize** menu eg. you can add a folder group on any [toolbar](#). If you want to change the folder tabs in a previously saved set, setup the active pane with the tabs you want, then save the set with the *same* name.

## Moveable panes



Except for the dual panes showing folder contents, all the other panes are auxiliary and optional. This includes toolbars and the preview, tree and other panes appearing in [Figure 1](#) (panes  — ). You turn them on and off using commands in **View** menu. All these panes share the same functionality, they can be resized, docked or floated, even minimized to a button, as will be shortly explained.

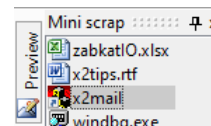
Let's start with the **toolbars**, including the addressbar. Toolbars can be rearranged with the mouse, if you drag them by their handle on the left edge, that looks like a dotted vertical line . You can put two or more toolbars in the same row, or split them into separate rows. Or right click on a toolbar and pick **Auto arrange** command from the context menu, if you don't mind the default widths. Other commands in the toolbar context menu:

- **Text labels** turn button labels on-off. You can turn off labels to save space; in such a case you can see the label as a tooltip window when you hover the mouse on a button.
- Tick **Small icons** to switch to smaller button images, again in the interest of saving space. Or tick the option off to see larger, more clearly identified icons.
- Once you are done arranging the layout use **Lock toolbars** to fix their position. The drag handles disappear. The only way you can move the toolbars is if you use the menu command again to unlock them (toggle)
- Toolbars can be [customized](#), so you can remove default buttons you don't need, and add the menu commands you use more frequently for easy access.



All peripheral panes (tree, quickviewer etc) can be rearranged to taste if you grab them by their titlebar and move them around to another side of the xplorer<sup>2</sup> window (all 4 sides are available for docking). An outline rectangle indicates the new docking location. You can have more than one pane docked to the same side. Using splitter bars you can resize both the entire dock side width (or *height* for top/bottom docks) and the relative sizes of panes within each dock. For clarifications please watch the [demo video](#)

If you hold down <CTRL> key while dragging a panel it will stay floating (detached). This way you can make it bigger without affecting the usable area of xplorer<sup>2</sup> window. Another space saving idea, especially for panes you don't need visible all the time, is to have them minimized. Use the little pin  on a pane's titlebar to auto-hide it. It then turns into a button (see **Preview** vertical button in the image to the right). If you hover your mouse over such a button, the minimized pane appears, taking up the entire side of the window. Move the mouse away and the pane will again hide. This is best understood with the [demo video](#) . The little **[x]** to the right of the titlebar will permanently close a panel. All these actions (float/dock/etc) are also available from a titlebar's context menu.

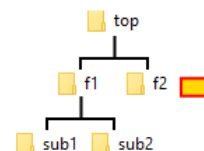



Once you setup the panes to your liking, use **View > Toolbars > Lock dock sites** command to fix their position. Note when you resize the entire xplorer<sup>2</sup> window, the dock widths and heights are adjusted proportionately, to keep the same relative sizes.

## Folder tree

Filesystems are organized like a **folder tree**, one folder can have subfolders, and these in turn can have sub-sub folders, and so on to greater tree depths. Folders on the same level are called *siblings*, and their direct subfolders are their *children*. Windows represents tree hierarchies in a more compact fashion using indentation to represent levels. Sibling folders are at the same (left) indentation, and their children are indented further to the right, as you can see in the pic to the right. Dotted vertical lines help identifying the branch levels.

Most people are familiar with tree controls, where the little arrows in front of each folder node are used to expand and collapse a sub-branch. If you prefer the keyboard, use up/down arrows to select a folder, then use right and left arrow keys to expand and collapse a branch, respectively.



- Normally the tree expands one level at a time. To expand **all branches** of the selected tree node, all the way down to the last subfolder, press <ALT+right> keys or use the forward toolbar button . Beware, expanding top level branches will take a long time! If you use this command by mistake press <ESC> key a few times and the read will be aborted.

xplorer<sup>2</sup> folder tree can be turned on/off using **View > Show tree** menu command, or pressing <CTRL+T> keys. It contains the entire filesystem starting with the **Desktop** root node. Once you find your desired folder, expanding tree branches as necessary, click on it and its contents will be read in the **active pane**. The selected folder will appear with **bold** letter in the tree. If you are a keyboard person, use the arrow keys to locate your folder. With the default *"single click to change folder"* [option](#) the last folder selected will be activated (there's a small timeout). If you want to browse the tree without activating folders, either disable the single click mode, or hold down <ALT> key as you move around the tree nodes.

- As the name suggests, the folder tree only shows folders, not files, but there's a loophole. ZIP archives and other compressed formats (7z, RAR) have a folder structure that can be browsed by xplorer<sup>2</sup>. If you want to see compressed folders and subfolders in the tree, untick the [advanced option](#) *"Sort zip folders as files"*

Except for choosing folders to browse, the tree can be set to follow the folder you browse in the active pane. This makes sense if you consider that you can switch to folders with all sorts of means in xplorer<sup>2</sup>, not just with the tree. This behavior is enabled ticking *"Keep synchronized with folder in active view pane"* [tree option](#). Then you can see the nearby siblings of the active folder. Without this setting, you can manually align the tree and folder pane using **View > Locate in tree** menu command.


By default xplorer<sup>2</sup> shows only a single tree that is shared among the dual folder panes. Some people prefer to have a separate tree for each pane, and this is yet another option called *"Show 2 trees"*. In this setup, the tree(s) become part of the dual panes and cannot be moved around the docks.

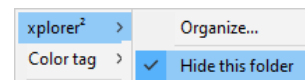


Each tree branch is arranged by Name, but it is also possible to sort sibling folders by date modified, using the background context menu, which displays when you right click on some empty space. With the same menu you can choose whether you want small or large folder icons (toggle). Finally you can turn the folder exclusion filter on/off with **Use blacklist** menu.

The entire tree namespace may include folders you never use, e.g. *Libraries* or some duplicate folders that appear in multiple locations. You can hide such unwanted folders by right clicking on them and then picking **xplorer<sup>2</sup> > Hide this folder** context menu command. Hiding a folder automatically enables the blacklist. If you want to see all the folders again (perhaps to *un-blacklist* some), pick **Use blacklist** from the background menu, to turn off the exclusion filter.

Formerly blacklisted folders have **Hide this folder** menu ticked, just untick the menu item to cancel the blacklisting. This command appears in the [customizable](#) xplorer<sup>2</sup> submenu, that is added on top of the standard shell context menu for files and folders. Note that only top level folders can be blacklisted.

For more information watch this [demo video](#) 



## Quick Access List

The tree holds all your folders and their subfolders in a huge and unwieldy structure. The Quick Access List (QAL) is a user configurable mini-tree where you put the folders you use the most. In fact it isn't even a tree, it is a flat structure that happens to be stored in the folder tree under **Quick access** node — which itself opens the system Recent items folder when clicked.

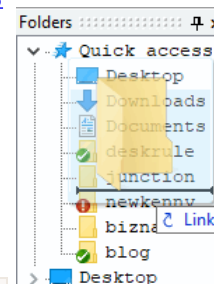
To begin with QAL is populated by your most frequently used folders, but you can add more folders with drag-drop (make sure you watch the [demo video](#)). Select a **single** folder either from the tree or a regular pane, and drag it over the QAL tree section. Be careful to drop it **inbetween** existing folders (you will see a horizontal insertion mark and the drop action will be **Link**), otherwise you'll end up moving the new folder inside the one existing in the QAL! For safety you can drag-drop new folders using the **right** mouse button, so if something goes wrong you can cancel the drop. Using drag-drop you can also reorganize items in the QAL, e.g. move them up or down.

Folders in the QAL can be used as bookmarks for browsing, [drop targets](#) for copy operations etc. The QAL branch can do everything the "normal" tree nodes do. There are only two exceptions:

- You can **rename** items in QAL without affecting the real folder name. This way you can provide more descriptive names and avoid duplicate folder names. Use the context menu or select a folder and press <F2> to rename it.
- When you press <DEL> key on a QAL folder, it merely *removes* said folder from the list, whereas if you tried the same in the "regular" tree, you'd actually *delete* the folder itself! To avoid mishaps, right click on the unwanted QAL item and use **Remove from pane** context menu command, which is clearer to understand.



If you remove QAL folders only to find them present again when you restart xplorer<sup>2</sup>, you must disable the feature that automatically adds the most frequently used folders in the QAL using [advanced options](#) (search for "Frequent items in tree")



If you don't need the QAL, untick "Show quick access list" [tree option](#). Then only the desktop-rooted folder hierarchy will show. Also note that QAL is not available when you opt for 2 trees.



Windows explorer also has a quick access list, but that is only related to xplorer<sup>2</sup> QAL by name. Adding or removing folders in windows explorer QAL has no effect to xplorer<sup>2</sup> QAL (and vice versa). Use drag drop to manage the QAL as explained above.

## Status information

The bottom line of xplorer<sup>2</sup> window is the status bar. It shows useful information about the program state, selected files and important messages. It is split up in several panes (up to 8, depending on your [settings](#)). Most panes show text information, and a couple show icons. If you hover your mouse pointer over a pane you will see a brief information about it in a **tooltip**, as below:

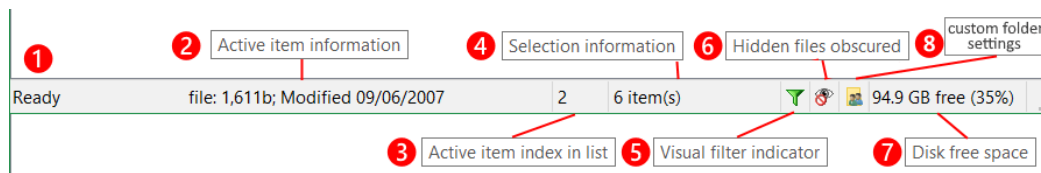




Figure 3. Status bar panes

- Status and messages.** The leftmost pane marked [1] shows the result of the last command (e.g. *10 items found*). If there was some problem, it shows an error message with a warning icon (and usually a **beep** sound)  **No item matches**. Such messages stay on for a few seconds and then they are dismissed automatically. Double click on the leftmost pane to repeat the last message (in case you didn't have enough time to see it); for extended error information use **Help > Last error** menu command.
 
 If you prefer error messages to stand out, tick the [advanced option](#) "Always make error messages a dialog...". Then whenever some error occurs, a modal message box will show up which you must acknowledge before you can resume with your work.
- Active item information.** The second pane [2] shows size and date information about the unique **focused** item (the one with the dotted rectangle around it). This is useful when you don't see file details in the folder (eg. if you use list view mode).
- Item index.** This pane is off by default but you can enable it through program options (**Tools > Options** menu, then **Advanced** property page). It shows the focused item index number in the list, beginning with 1 (first item), 2 for the second and so on.
- Selection information.** When nothing is selected this pane shows how many items are in the folder altogether. Once you start selecting 2 or more items, this pane shows how many items are selected and their total size e.g. **2: 1,393,455** means there are 2 files selected and their total combined size is 1393455 bytes.
- Visual filter indicator.** When a [visual filter](#) (e.g. a wildcard) is active, obscuring some folder items, a green filter icon appears here; when there's no filtering involved (all items show normally) pane [5] is blank. You can double click on this status bar pane to toggle the visual filter on/off
- Hidden files indicator.** Special files like *DESKTOP.INI* and such with **System** or **Hidden** attributes, can be filtered out in bulk if you tick off "Show hidden files" option. In this case the eye icon will appear in this pane indicating some files may be hidden from view. Double click on this pane to toggle hidden files mode on/off.
- Disk free space.** The rightmost pane shows the free space in the current hard disk (e.g. if you are browsing *C:\TEMP* folder it will show the free space in *C:\* drive), both as an absolute GB magnitude and a percentage. To see free space for both left & right pane disks simultaneously, use **Tools > Free space** menu — this will show in the leftmost pane.
- Custom folder indicator.** An icon appears at [8] when you browse folders with special view modes (saved [folder settings](#)). For regular folders this pane is blank.

But the status bar usefulness doesn't stop here. If you browse the menu commands, you will see a brief explanation of what each command does on the statusbar! This also applies when you hover your mouse pointer over a toolbar button. It is a mini instruction manual. Often while xplorer<sup>2</sup> is busy executing one of your commands, a **progress bar** shows on the statusbar, giving you an idea how much more is left to be done.

Not bad for a little bottom line, isn't it? If nevertheless you don't want any of the above, you can disable the statusbar altogether using **View > Toolbars > Status bar** menu command. If you disable the statusbar, all error messages will show as message boxes.

 Status information is taken from the active folder, wherever it may be (left or right dual pane). If you prefer a **separate** mini-status bar for each pane, turn on infobars (**View > Toolbars > Info bars** menu command). You can customize infobar panes using the right click context menu, to show any piece of file information available.

## Ribbon toolbar

The ribbon user interface combines the menu bar and toolbar in a tabbed multiline super toolbar UI. It is simpler and groups commands better in tabs and groups within each tab. Use **View > Ribbon** menu command to turn on the ribbon interface. This will automatically hide the menu bar and toolbar (compare with [figure 1](#))

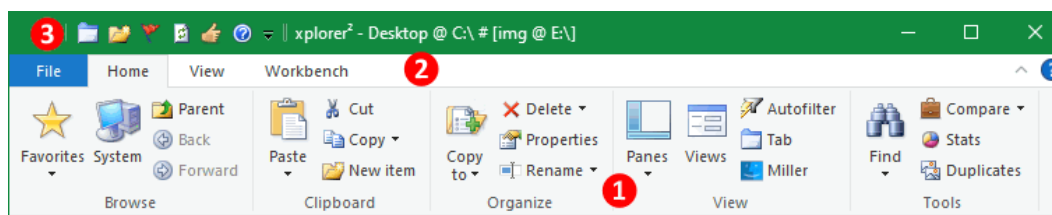



Figure 4. Ribbon toolbar interface

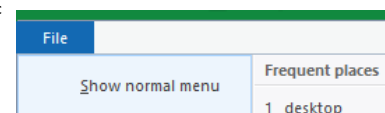
The ribbon has 3 command tabs (item ② in the above figure). The basic **Home** tab contains the most frequently used commands, whereas other tabs offer more detailed control for advanced features. The leftmost **File** tab is a traditional drop-down menu where you can exit the program, get help, and get back the original menu if you don't like the ribbon!


The command buttons in each tab are split in categories (see ① Browse, Clipboard, etc), or logical groups. Many ribbon buttons are split into a main command like Paste and a drop-down portion that shows variations of the main command (e.g. Paste link). You can get information on what each button does if you hover the mouse over it.

The ribbon offers limited customization support. You can right click on a button you use frequently and put it on the Quick Access Toolbar ③, the little toolbar the ribbon puts on the window titlebar. You can put up there all the commands you need then right click on the ribbon to minimize it, and reclaim the space it uses.

Most of the commands in the full menu system appear in the ribbon, albeit at different, more intuitive locations. This is a basic choice you have to make early, because if you learn the ribbon version of the commands it will be hard to revert to the menu bar interface (and vice versa).

If you want to get the menu bar back, you can remove the ribbon using **Show normal menu** command under the leftmost File ribbon tab. Or if you just want to use an advanced command that's not available in the ribbon, use the [command finder](#), the little question mark button  on the quick access toolbar.





 Most xplorer<sup>2</sup> users opt for the traditional menu bar instead of the ribbon UI. Therefore most of this help file is talking about the menu version of the user interface. If you want to use the ribbon and cannot find a particular command described in the help, use the command finder with the exact command name. Also try to use the **context menu** whenever possible, most UI elements support them with most relevant commands.

## Typical workflow experience

When you first install the program you'd probably need to decide how xplorer<sup>2</sup> window looks like, how big it is, which panes it shows, so that the layout feels comfortable for your style and needs. In fact you can save multiple window [layouts](#), one for each type of task you perform. Other major decisions at this stage are whether to use the [ribbon](#) or a normal menu, and whether to opt for [dark mode](#) user interface.

Your everyday interaction with xplorer<sup>2</sup> is to [browse](#) one or more folders, then click on files to manage them this way or the other. xplorer<sup>2</sup> helps you get to your folders quickly, and ultimately find the files you are after, with pin-point accuracy. Then use menus and toolbars to work with your files. Most of the time you work in one folder at a time, as your filesystem is organized in your drives. Using [scrap](#) windows you can work with items from many folders simultaneously — e.g. modify files in a folder and all its subfolders.


Working with the mouse is easy, but for increased productivity xplorer<sup>2</sup> is **keyboard friendly**. Almost everything can be done with the keyboard. Lots of commands have keyboard shortcuts; if you are used to different accelerator keys, you can [remap keys](#) to your taste. You will find reminders of keyboard shortcuts in menus **Find files...** **Ctrl+F** and toolbar infotips  so you know that pressing <CTRL+F> is a quicker way to access the file search command! To move around the various panes, there are many keyboard shortcuts that involve the <TAB> key, e.g. <SHIFT+TAB> jumps to the addressbar. There are more useful [hidden](#) keyboard shortcuts in the appendix


 When browsing menus and dialog boxes, look out for underlined letters e.g. **B**rowse (you may hold down <ALT> key to see the underlines). For menus just press the underlined letter (e.g. B) and the respective command will execute. In dialog boxes, you can jump quickly to an input field using its underlined letter pressing <ALT+B>. In dialogs, pressing <ENTER> is like clicking on **OK** button and to cancel press <ESC>.

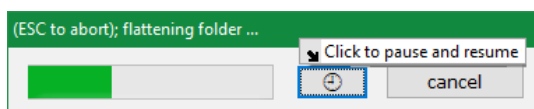
Here are some concepts to keep in mind for your everyday file management:

- **Focus.** Despite the apparent complexity of the user interface, there is only one active folder at a time, the pane where you last clicked in. In this active folder, there is only one item that is the **focus**, again the one you last clicked on. All information panes, previews etc show information about this special file or folder item.
- **Selection.** Using simple clicking or any [precise selection](#) methods, you choose one or more items to operate on simultaneously. All menu commands work with the selected items in the active pane. Note the focused item is usually also selected.


xplorer<sup>2</sup> is built to be **fast**, efficient and responsive, even when you work in large folders with thousands of files and complex properties. When it needs to extract slow information, e.g. a movie duration, it does so in a background thread. When you browse a big folder, you can start working with files **immediately** even though it may seem that the reading job is only half done. xplorer<sup>2</sup> is busy fetching information in the background and will present it whenever it becomes available. File icons are extracted last of all, because they are not essential to file management.

Most menu commands will open a **dialog box** that lets you setup the file operation to taste. All these dialogs are described in detail in this documentation. All of them include an **OK** button you use to start the command procedure, and **CANCEL** which you click to abort the operation before it even begins — in case you picked the wrong menu by accident. Some dialogs are **resizable**, look out for the sizing handle  at the lower right corner. To help you focus on the active dialog, xplorer<sup>2</sup> **shades** (darkens) its main window till you finish setting up the dialog's parameters. If you find this behaviour distracting tick the [advanced option](#) "Disable main window alpha shading..."

When xplorer<sup>2</sup> is really busy working on your commands, especially if the job is big and hard, then you need to wait a while (a wait cursor  will show). Most lengthy operations show a **progress** dialog that lets you see how much is left to be done, as well as buttons to **pause** or cancel the job altogether, as such:




Some less intensive jobs may show a progressbar on the [statusbar](#) instead, which again gives you an idea how much work is left to be done. To cancel such commands press <ESC> key (if it doesn't respond immediately, try keeping ESCape pressed down for a few seconds).

 Some jobs need time, and there's not much you can do about it. Usually xplorer<sup>2</sup> doesn't require any response from you during the work, so you can just leave it unattended. If you want to keep on working with other files and folders, you can use **Window > Clone** menu to open another xplorer<sup>2</sup> window (but do so *before* you



get busy), or start another xplorer<sup>2</sup> instance from its desktop icon. xplorer<sup>2</sup> is designed to work with multiple windows in a single process, being gentle with your system resources.

During file operations, it may happen that folder contents get changed, files added or deleted or modified. xplorer<sup>2</sup> senses such changes automatically and refreshes the active folder pane so it stays up to date. This feature is working most of the time but if you find that xplorer<sup>2</sup> didn't pick up all the changes use **View > Refresh** menu command to force a re-read of the contents. Working inside a constantly modified folder (e.g. Downloads folder during an active file download) may waste too many resources, so you should consider suspending the autorefresh feature using **View > Hold autorefresh** menu item.

 Autorefresh may cause folder contents to "jump", xplorer<sup>2</sup> scrolling the folder pane away of the file you are trying to work with — because xplorer<sup>2</sup> tries to bring the focused item back into view after an automatic refresh. To avoid such view jumps, either click on the item you are working with (make it focused), so xplorer<sup>2</sup> won't scroll away, or disable the program [option](#) "Automatically re-sort contents..."


During file operations, problems are possible for a variety of reasons and causes. In case of **errors**, xplorer<sup>2</sup> will pause what it's doing and alert you to the problem. Some errors can be ignored or skipped, but it depends on the command being executed, the folder type and so on. Some errors are terminal, so xplorer<sup>2</sup> will abort the operation and issue a warning message on the status bar (you will hear the dreaded ding sound as well).

In general errors are caused by inadequate **permissions**, e.g. you are trying to copy into a folder that belongs to another user. Sometimes you can work around UAC restrictions launching an elevated xplorer<sup>2</sup> process via **Window > Administrator** menu command. However it is not recommended to run in administrator mode all the time. You can tell when xplorer<sup>2</sup> is running elevated by the \* showing after the program name in the window titlebar. **Window > Administrator** command is then *disabled*.

See the [troubleshooting](#) section for some common problems and workarounds.

xplorer<sup>2</sup> is well tested and **stable**, so it shouldn't crash or hang very often. There are thousands of active users so real problems get reported and fixed quickly. However you cannot expect taxes and program death <g> so crashes are possible, especially if you use buggy shell extensions. If you find a reproducible bug please report it using **Help > Crash information** menu command. See the [appendix](#) for the kind of information required to troubleshoot crashes and hangs.

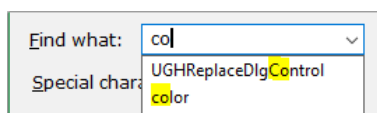
Under normal circumstances, xplorer<sup>2</sup> automatically saves its settings on exit, which means next time you run it, it should look exactly like last time you used it (window size and position, folder tabs, the lot). If you want to start xplorer<sup>2</sup> at fixed locations, use [command line](#) arguments to specify fixed folder paths for the left and right panes.

 When xplorer<sup>2</sup> cannot re-establish connections to folders you browsed last time you used it — e.g. disconnected network drives, some tabs may end up browsing Desktop instead. If you reconnect those remote folders, use **Customize > Folder groups > Startup** menu to restore the original tab locations.


## Input history

Most user interface elements that accept typed input, maintain a history of past inputs (30 items by default). This history is available from the drop-down portion of combo-box controls — the most frequently used UI element. So next time you use some command, you can scan past inputs and quickly reuse them avoiding extra typing.

To access this history, either click on the drop-down arrow button of the combo-box, or start typing and xplorer<sup>2</sup> will find all the matching items from the history and present them to you as such:



Notice that the partial string you type is matched both at the beginning or the middle of history strings. Once you see the item you need to repeat, press <Down> arrow a few times to select it, then <ENTER> key to close the autocompletion popup. Press <ESC> key to close the popup without selecting anything.

 Some people are overly sensitive with **privacy**, even when it comes to their own data. If you don't want xplorer<sup>2</sup> to remember past inputs (or folders browsed), either turn off "Save program state on exit" [option](#), or set "History items maintained..." option equal to the special value 2, which means no history is remembered.

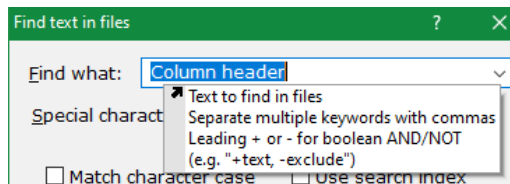
## How to get help

xplorer<sup>2</sup> can do many things for you, so when you reach for advanced commands beyond simple drag and drop, things can get complicated. However there's good help to guide you, right inside the program:

- The [statusbar](#) shows brief explanations as you scroll through the menu commands or fly your mouse over toolbar buttons.

Add current folder in the bookmark list

- Many dialog box controls show a help tooltip to describe their function, especially the more complicated ones. Leave the mouse hovering over a dialog control and the explanatory tooltip (balloon help) will appear as such:



- Most command dialog boxes are linked to this help file. Press <F1> key (or the question mark [?] on the dialog's titlebar right end) and the help for the current dialog will be loaded, where you will find detailed explanations.
- Use **Help > How do I?** menu to open the onboard demo video panel. Press <CTRL+F> keys to search within the video panel for the function you need. It is recommended you watched the first few demo videos, they will teach you the basics and give you a quick start using xplorer<sup>2</sup>
- Use **Help > Online support** menu to send an email to xplorer<sup>2</sup> support team. If you have any question, don't hesitate to ask!

There are also many **online** help resources you can use:

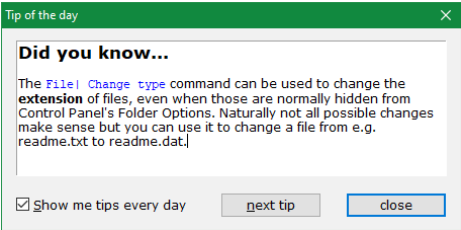
- The Frequently Asked Questions ([FAQ](#)) page
- The xplorer<sup>2</sup> [forum](#) where you ask questions and search for previous answers
- Our [facebook](#) and [twitter](#) pages
- zabkat's [youtube channel](#) for more demo videos
- Search the [blog](#) for past articles explaining xplorer<sup>2</sup> and windows file management in general

Use **Help > Tip of the day** to see bitesize, helpful tips and tricks every time you start xplorer<sup>2</sup>. The tips window is modeless, meaning you can read the tip and try it out at the same time. When you had enough of these tips, untick "Show me tips every day" checkbox — but leave them on for a while, you will be surprised by what there's to learn!

All the tips are available in a single file called X2TIPS.RTF in xplorer<sup>2</sup> installation folder, if you prefer to search in them, print them out etc.

Further reading

The most useful xplorer<sup>2</sup> tips and tricks for [better everyday](#) file management



From our experience with customer support, some commands tend to confuse users. In such cases xplorer<sup>2</sup> shows a 1-off clarifying message. If you don't want to see that message again, tick **Don't show this message again** and you won't be bothered with it again.

Command finder ▶ PLAY

If you cannot remember where's the menu command to perform some action, use **Help > Command finder** menu command. That's the only menu command you need to remember! A search dialog will appear where you can use one or more keywords to find matching commands, and you should be able to find what you're after quickly.

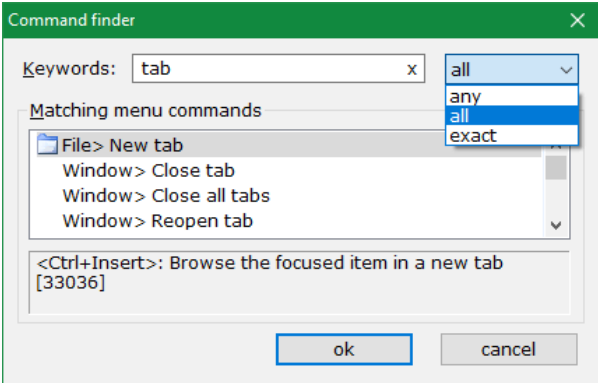



Figure 5. Command finder dialog


The command search algorithm uses menu text and the brief explanation messages that appear on the statusbar describing each command — so you don't need to remember the menu name, you can search for what the command does, too. Use the **Keywords** input box for your search keywords; any matching menu commands will be listed in the results box. Note searches are case **insensitive**, so you can use capitals or lowercase keywords. If your query returned too many results or no results, try changing your search keywords.

If you are using a [translation](#) of xplorer<sup>2</sup> GUI, you must use the same language for search keywords. You search for commands as they appear in the menu bar — but you don't need to worry about accented letters: *rôle* is the same as *role*.



If you search with 2 or more keywords, you can specify the **context** from the drop down list next to the keyword box. Use **All** to find all your keywords (in any order); this option combines accuracy with flexibility ("tab new" will find **New tab** command). **Exact** option will only match whole phrases ("tab new" will *not* find **New tab** command). Finally use **Any** match option to find commands that contain any one of your search keywords. This will return many results, perhaps *too many* — but it is useful when you cannot find a command any other way.

The search results box lists all commands that match your query, one command per line. The full "menu path" is presented e.g. **File > New tab** means open **File** from the menu bar and use **New tab** command from File menu. Some commands may have 3 or more segments if they are in submenus (the **>** character separates submenus). However you don't need to remember the whereabouts of the menu, just **double click** on a command in the results pane and you immediately execute the command — just as if you used the menu! If you prefer the keyboard, select the command row and press <ENTER> key to execute it.




The dialog shows the quick description for each matching command, including the accelerator key (<Ctrl+Ins> in this example), and the actual **command ID** number (33036 for new tab). This number is handy for writing [macros](#)

## Browsing the shell namespace ▶ PLAY

If you only had a few files to deal with, you wouldn't need folders. Like some ancient MSDOS versions, you would just put all your files in one folder (aka *directory*) and use their name to tell them apart. But if you have thousands of files, some documents, photos, movies and all the rest of it, you need to hold them in separate folders to keep some order. A **folder** is the equivalent of a desk drawer, where you keep your stuff organized. But folders are super-drawers, where you can have folders within folders for finer levels of organization, and subfolders inside those, to a potentially infinite degree. And that's just your own documents, your computer needs tons of system files to operate, you install extra programs, and all these add to your files and folders. The entire hierarchical tree of folders within folders is called the **shell namespace**.

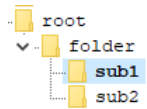
You can create folders at will, move them around, copy and delete them. Likewise you can create, move, rename and delete files within any folder. There is only one rule: each item in a folder must have a **unique name** — but you can have identical names in different folders.

 The folder tree idea has been around for ages. Over the years there have been attempts to replace it with "easier" organization methods like tag management and databases, but manual organization with folders has stood the test of time.


In the old days of DOS consoles (DIR command anyone?) folders had to be reached by typing their names. To browse a folder a few levels deep, you had to type the names of all intervening folders, the **path** you had to travel to reach your target folder. The full path is the address of each folder starting from the top (root) folder. As at each level names are guaranteed to be unique, the path can identify any folder unambiguously.

Paths use the **backslash** \ special character to separate folder names. The backslash is a reserved symbol that isn't allowed to be used in folder or file names. For example to address the subfolder in the pic to the right, we compose its path using all higher (parent) folder names, starting from root as such:



```
root\folder\sub1
```



Files also have path names, you just append the file name to its parent folder path name, using a backslash as a separator as usual. Thanks to GUI programs like xplorer<sup>2</sup> you no longer need to type folder paths manually, but you come across them all the time so it's good to know what they are.

 Try to keep your folder names as short as possible. Don't go overboard with verbose descriptive names. Windows filesystem has a limit of 255 letters for the overall path name, so if your names are too long you won't be able to create deep folder hierarchies. xplorer<sup>2</sup> can handle [deep paths](#), but most other programs can't, so try to avoid them. Filenames on the other hand can be more descriptive if you like.

With xplorer<sup>2</sup> you seldom need to type folder path names manually. A large proportion of the program's infrastructure is there to help you get from A to B quickly and effortlessly with a variety of folder access mechanisms:

- Using the [folder tree](#) pane and quick access list
- Typing paths in the [addressbar](#) or revisiting a folder from the addressbar's history
- Back/Forward chain navigation using <Alt+ArrowKeys> (or **Go to** menu)
- <Backspace> key takes you **Up a level** to the parent folder. You can also enable the traditional UPDIR  item using [advanced option](#) "Show UPDIR item". Double-clicking on some empty space (on the active pane's background) also takes you one level up
- [Bookmarks](#) (favorites) menu
- Frequently visited folders are automatically remembered and accessible from **Go to > Frequent** submenu.
- Frequent folders also appear in the **jump list** (right click on the xplorer<sup>2</sup> taskbar icon)
- Reinstate a previously saved folder group (**Customize > Folder groups** menu) as a set of folder tabs
- A variety of keyboard shortcuts (e.g. <Ctrl+Shift+D> will take you to D:\ root drive)
- Select a local/mapped drive using the  toolbar button; the same drives appear as a drop-down menu from [ThisPC](#) button on each pane's titlebar.
- Access any local or mapped drive using the drivebar (UI element [6] in [figure 1](#))
- Standard system folders like the control panel, my computer etc are easily accessible from **Go to > Special folders** menu.
- Using the titlebar of each pane: right-click to access a list of recent folders, or click on a subpath (see **Figure 6**). If parts of the path are cropped or obscured due to width limitations, press <Shift> while right-clicking to get the subpaths in a menu.

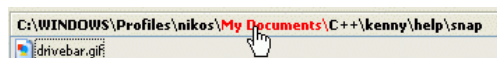



Figure 6. Selecting a subpath from a pane's titlebar

Normally you browse folders in the active pane but there are other possibilities, depending on keyboard modifiers:

- If you hold <Alt> as you are changing folder, it will be loaded in the *inactive* pane.
- If you hold <Ctrl> a new window will be **cloned** and the folder will be browsed there.
- **File > New tab** menu will open the folder in a new tab, part of the active pane.
- Likewise **File > Column mode** will open the selected folder item in a new [miller pane](#) showing side by side.

 Installing a generic [WCX](#) plugin like [Total7zip](#) allows you to browse into "hidden" archive formats, e.g. APPX, JAR or DOCX, even NSIS installers (like xplorer<sup>2</sup>\_setup.EXE!). Use **File > Enter archive** menu to gain entry to such "folders"; once inside browse their subfolders with <ENTER> key as usual.

## The addressbar ▶ PLAY

Element [5] of the main window is where you type paths to access folders by name, if you are old school. It understands both full and **relative** paths, e.g type a subfolder's (plain) name to enter it — as long as you are browsing the parent folder. After you type a folder path, press <ENTER> key or click on the little green arrow to browse the folder in the active pane:

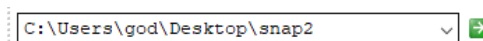


Figure 7. Addressbar combo box

You can also type **file** names (with or without a folder path), which is a quick way to browse a certain folder and select the named file item in one stroke. It also accepts simple [wildcards](#) that constrain item visibility.

The addressbar isn't clickable; for this functionality use the pane's titlebar (see [figure 6](#)). Remember xplorer<sup>2</sup> introduced crumb browsing years before windows explorer copied the original feature. To save you typing it supports path **autocompletion**. If you tick "Explorer-style path autocompletion..." [option](#) it helps you complete path segments using a popup completion list.

The combo box drop-down portion includes a recent history of your browsed folders, saved bookmarks and your most frequently used paths. You don't need to click on the

rightmost arrow extend the drop-down history, just type *part* of the path you are after and hit <Up-Arrow> key to fetch the first matching path; if that isn't the one you are after, press <Up-Arrow> again a few times, to see all completions. For example, typing **f**resh would match a history folder named C:\TEMP\FRESHMAN. You can also type saved **bookmark** names.

You can also start [external programs](#) and pass files in the active pane as command line arguments. Type a > prompt character followed by a program name to run it. For example type:

```
> NOTEPAD $N
```

and press <ENTER> to start notepad and edit the currently focused file in the active pane!


\$N represents the selected filename and is one of the available [tokens](#) that let you build a reusable command palette.

## Folder types


Most of the folders you are working with are of the "regular" variety, kept on a permanent storage medium like a hard disk or USB stick. These are called **filesystem** folders, and although they have a few subtle differences (NTFS or FAT32 formatting) for most everyday usage they are all the same. Local drives connected to your computer are identified by a **drive letter** (customarily the main hard disk is called C: and contains your windows installation). Each extra local storage medium has a unique drive letter.

Another class of filesystem folders are remote or **network** folders, which give you access to hard disks of other computers via LAN or other connection. Instead of a drive letter, network roots are identified by a PC name and a root share folder, with two leading backslashes:

```
\\PCName\share\subfolder
```


 It is possible to map a network drive to a drive letter like Z: using **Tools > Map network drive** menu command. Mapping saves you typing long UNC paths; instead of the full share name you would type Z:\subfolder

Network folders are much slower and xplorer<sup>2</sup> takes special [precautions](#) when dealing with them. Other remote access folders are **cloud** drives like OneDrive, DropBox etc, but they also behave like regular folders most of the time — except for [cloud-only](#) files that are available on demand. xplorer<sup>2</sup> is careful not to cause unnecessary downloads for such cloud content.

 NTFS supports **reparse points**, which allow folder shortcuts (junctions, symbolic links, mount points etc). They allow you reorganize your filesystem, e.g. you can turn two unrelated folders into siblings, like as if they belonged to the same parent folder. Junctions are created by [special paste](#) commands.


Despite these local/remote differences, all these folders are regular filesystem, and you manage them in a similar fashion. All xplorer<sup>2</sup> menu commands work in filesystem folders.

Another folder class are in reality files, **compressed** archives like ZIP and RAR. They contain a local filesystem of files and subfolders, that's why windows shell treats them as "folders". Only a subset of commands apply to zipfolders. They are also slower than normal folders, as their contents need to decompress for many operations. As they contain files, xplorer<sup>2</sup> tries to offer a near-filesystem experience in terms of browsing and previewing.


 Browsing compressed archives as folders is useful in many respects, you can search and preview items in zipfolders, compare them with regular folders etc. However some people prefer using the dedicated program (WinZip, 7zip ...) instead of xplorer<sup>2</sup> opening them as folders. To disable ZIP or any type of compressed folder extension, add them as a comma separated list in **"Blocked archives"** [advanced option](#). On the other hand if you want to browse into *more* compressed archives (7z/RAR/TAR etc) as folders inside xplorer<sup>2</sup> install this [shell extension](#).

xplorer<sup>2</sup> can also browse mobile phones (cells) and cameras and extract your photos, videos and other files. When you connect your phone via USB cable, it will appear as a special folder in ThisPC and you can browse into it. Such devices aren't read as normal filesystem folders, they are based on MTP protocol, and are quite slow to access. You are better off removing the SD card from your phone and using a card reader to get your photos faster.

Finally there are virtual folders, shell constructs that merely organize real folders, like ThisPC that contains all local and connected hard disks, Recycle Bin that manages files deleted and a few more special folders. You don't do much file management inside such folders, and each has its own peculiarities. Menu commands that are not relevant in special folders are **disabled**.

 Special and virtual folders have weird "paths" like ::{20D04FE0-3AEA-1069-A2D8-08002B30309D} which corresponds to ThisPC folder. It is unlikely you will remember these cryptic numbers but they are nevertheless valid identifiers, and can be used to set bookmarks to special folders.

**Further reading**

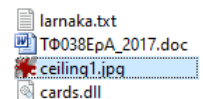
- Some [uncommon](#) folder paths
- Browsing into compressed zipfolders [demo video](#) 


## File types


All folders are the same kind of object, but files come in many varieties. You have plain text files, images, office documents, music, the list is very long. Windows shell uses a simple method to discover file types, by filename **extension**. This is (usually) a 3 letter type abbreviation like TXT for plain text, JPG and GIF for images, PDF and DOC for office files and so on. This extension is added past the "real" (base) filename with a separating dot (e.g. PLAIN.TXT). When you try to open a file, windows looks up the program that is associated with the filename extension and starts the registered program to read your file.


File type is also associated to an **icon**, the user friendly way to tell file types apart. All text .TXT files have the same icon as in the pic to the right. These icons are the only way to tell the kind of files if you disabled extensions (ticked **"Hide extensions for known file types"** system folder option).


Whereas you are free to rename the base name portion of a filename to help you identify it, it is unwise to change its extension. This poor attempt at file type conversion will probably make the file unreadable. Only plain text files can have their extensions changed without risk.



 Unlike files, most folders have the same bland yellow icon. You can make a folder stand out giving it a different icon, using its [property page](#), then switch to **Customize** tab and click on **Change icon** button. You can even assign custom icons for [external disks](#)

Shell also uses icon **overlays**, little modifiers of basic icons, to convey extra information about a specific file. For example version control systems like TSVN use overlays to indicate the modification status of files in a repository. Another common overlay is the little arrow icon that denotes a shortcut file, a special file type with .LNK extension that is a "pointer" to a real file located in another folder (like the program shortcuts  on the desktop).

 File icons are pretty but consume more system resources than you can imagine — especially so custom icon overlays from shell extensions. If you don't care much about icons and overlays, you can tell xplorer<sup>2</sup> to use simpler icons from program [options](#), or even don't show icons at all and pretend you are back in 1989 <g>)

 Back in the DOS days, filenames couldn't contain spaces or special symbols, and they were limited to 8 letters maximum, plus a 3 letter extension, the so called **8.3** naming convention. Modern shell allows filenames up to 255 letters long with only a few prohibited symbols (you can't have \: and such path separators). Some hard disk partitions (e.g. C:\) have both long and equivalent 8.3 filenames (e.g. a file like JAZZ HOUSE MIX SESSION.mp3 can also be addressed as JAZZHO~1.MP3). xplorer<sup>2</sup> can work with long or 8.3 names just as easily. When filenames contain spaces, you must enclose them in "quotes" to pass them as command line arguments — or use their 8.3 equivalents, if available.

## Folder bookmarks

There are thousands of folders under the desktop root tree item, but you only work with a few of them most of the time. Using bookmarks you can quickly locate these most frequently used folders and increase your productivity.

One way to organize bookmarks is the **Bookmarks** menu that looks like the pic to the right. It is a flat list of folders with a customizable icon and an optional keyboard shortcut for keyboard enthusiasts. You add folders to the list using **Add current** menu, which adds the current folder in the list with a name you choose. Then choose a folder from this menu list to browse it instantly.

As you add more folder bookmarks, this list will grow and cannot be seen in its entirety in the menu. Any extra bookmarks will be available from **Bookmarks > More** menu, which opens a dialog that shows all the saved folders (looks similar to figure 8 below), for easy picking. By default up to 25 bookmarks will show in the menu; to see more increase the "Custom menu items" [advanced option](#), a setting that affects other similar menu lists (under **Customize** menu). Up to 99 bookmarks can be saved in this fashion.

Bookmarks, like any menu items with icons, can be added to the main toolbar (item [7] in [figure 1](#)) for easier access. Right click on the toolbar and pick **Customize** from the context menu to add one or more bookmarks. Watch this [demo video](#) for more details. Bookmark names can also be typed in the [addressbar](#) and also autocompleted by partial name and <Up-Arrow> key.

A bookmark stores just one folder that is loaded in the active pane. **Bookmarks > Dual** submenu manages a similar list of **paired** bookmarks. If you are in dual pane mode, **Bookmarks > Dual > Add current** will store both left and right folders as a pair; when you reinstate such a pair it will bring up two folders instead of just one. Another menu list caters for sets of tabs: **Customize > Folder groups** submenu will save all the tabs in the active pane as a single item. If you pick a saved folder group from the menu, all the folder tabs will be recreated in one stroke. All these menu lists are managed in the same way.

As you traverse the menu items of any customized list, you get to see the "description" of each item on the statusbar, so you can e.g. see the path that each bookmark menu item represents: `Go to this folder [C:\VirtualBox VMs\win98-import]`

Note: in the [ribbon](#) interface bookmarks are called **favorites**, and a similarly operated menu drops under the star ★ button in Home tab.

**Bookmarks > Organize** menu command lets you manage the menu list. Organize dialog shows all the stored bookmarks, including any that are not seen in the menu (if there are too many). This dialog allows you to **reorder** items (put frequently used bookmarks further up in the list), change properties and **delete** bookmarks that are no longer necessary.

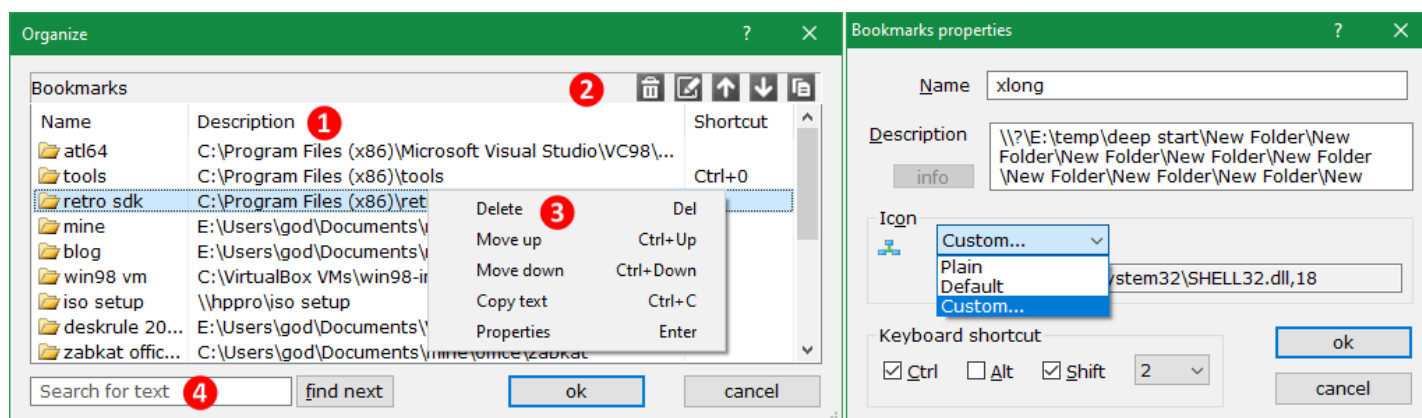


Figure 8. Organize bookmarks and change properties dialogs

The organize list shows bookmark details [1]: the name, "description" (this is the actual folder path), and any shortcut key associated to the bookmark for easy invocation by keyboard. To operate on the list use the little toolbar [2] (use the button popup tooltips to see what each one does) or equivalently right click on an item to modify it using the context menu [3]. Use the search box [4] to find a bookmark by name. This box matches strings in both name and description columns. Click **Find next** to find the bookmark; if that wasn't the one you wanted, click **Find next** again for the next match.

Double click on an item to modify its properties; a secondary dialog will open that lets you change the Name, Path, select a custom icon and assign accelerator keys (they must be a combination of <CTRL> and <SHIFT> with a number 0-9. Don't use <ALT> because it will make the bookmark load the folder in the *inactive* pane.

The default bookmark icon is that of the bookmarked folder — that most of the time looks the same. Use **Custom** from the drop-down list to choose a completely different icon, that helps you find the bookmark easier (especially if you add it to a toolbar). The **Plain** icon option assigns a plain but fast folder icon, which is advisable for slow network folders. When you are done changing properties, click OK to confirm or cancel to discard any changes.

You can also set a bookmark on a **file**. This will browse the container folder and select the file, ready for action. First set the bookmark on the parent folder as usual, then use the bookmark properties above to add the file name in the "description" field!

Managing and using bookmarks from a menu is not the most efficient way to go around browsing favorite folders. You can also use the tree's [quick access list](#) and the [mini scrap](#) pane to drag-drop folders, then click on them to browse them.

There is also the legacy *quick bookmark* in **Go to** menu, intended for temporary bookmarking. You set it on the current folder browsed using **Set quick-mark** menu, then you can quickly return to the same folder using **Quick bookmark** menu or by <CTRL+F1> keys. There are tons of alternative ways to browse a favorite folder in xplorer<sup>2</sup> — add this extra one in the bargain :)

#### Further reading

- How to organize bookmarks into [subfolders](#) per theme

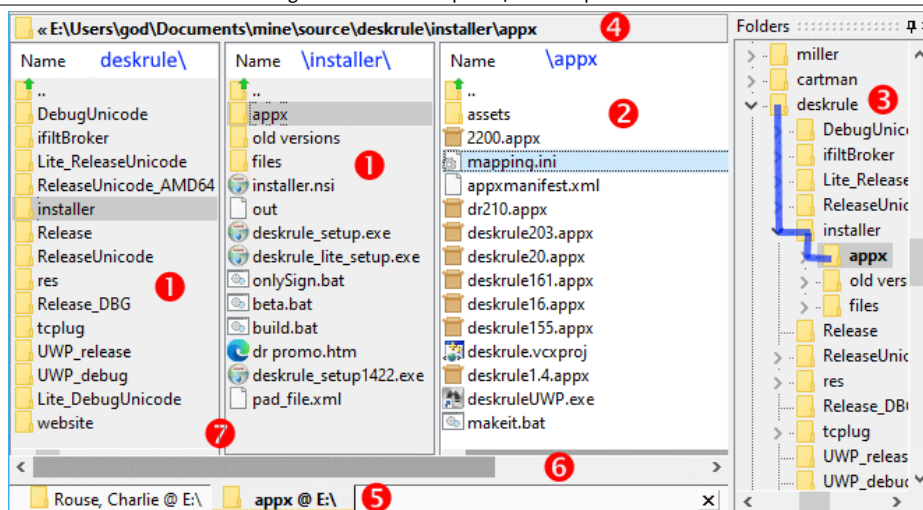
## Column mode

Usually when you browse into a subfolder, the current tab switches entirely to it. Say you were browsing C:\path\DESKRULE and then you entered the subfolder C:\path\DESKRULE\INSTALLER, you'd expect to see the contents of **INSTALLER** folder replacing the old pane contents. If you turn on Miller browsing (**File > Column mode** menu) then **both** the original and the subfolder contents will show side by side. If you enter a new sub-sub folder e.g. C:\path\DESKRULE\INSTALLER\APPX then a third side panel will open, and so on. As you browse deeper, panes to the left shrink in width. If you have too many folders simultaneously open, a scroll bar will appear to help you manage the panes, since only a few of them will show at a time.

Miller browsing will strike Windows users as odd, but it is popular with Mac computers and their **Finder** file manager. Its advantage is that you can see neighboring folders simultaneously and transfer files from one to the other. Like dual pane browsing, only increased to multiple panes!

You move from one miller pane to the next either with left/right arrow keys, or clicking with the mouse. Only one folder is **active** in a miller column set, which will show in white background; all the other folders show with a darker background. The active folder is dominant: its path shows as the tab's current path, and all menu commands will operate in files selected in the active pane, just as if it was the only folder showing. After you create miller panels, each one could be set independently (e.g. one in thumbnails, the other in details, and a 3rd in list mode. But usually they all share the same view mode.






**Figure 9.** Side by side Miller browsing

Let's go through the elements of a tab showing side-by-side folders, showing this mini-hierarchy:

DESKRULE  
INSTALLER  
APPX

- **Inactive folder panes [1].** These are folders higher up, browsed earlier. They show with a darker background. If you click in an inactive pane then it will become active, deactivating the previous focus.
- **Active folder [2].** This is where you work in, selecting items to open, copy or preview. It shows with a lighter background color. Use up/down arrow keys (or PageUp/Down) to move within the active folder's contents.
- **Folder tree [3].** The tree isn't a part of column browsing mode, but it is shown here to aid your understanding the relationship between the side-by-side panels
- **Pane's titlebar [4].** Shows the path of the currently active folder
- **Tab strip [5].** The title of the active tab represents the currently active folder in the miller set. You can combine multiple tabs with miller mode, and each tab can show a different set of neighboring folders.
- **Main scrollbar [6].** When you have too many side-by-side folders open, there's no room to see all of them, that's why a scroller pops up to help you manage the folders. Each individual folder pane is also scrollable, but by default its scrollbars show in miniature, to save on screen space.
- **Splitter bar [7].** Vertical divider bars separate miller columns. You can resize each pane dragging its splitter bar with your mouse

When you open a new miller pane, the old pane shrinks down in width. By how much? If the pane is in detailed view mode, then it will be shrunk to the width of the NAME column. If it is in list or thumbnail mode, then it will shrink to the size of the currently selected folder item. You can set a maximum width limit to miller columns from program [options](#) (Window property page) setting a value for *Max list column width*. There are more tweaks available using **Tools > Advanced options** editor, e.g. the maximum number of side panels (default is up to 6 parallel folders).

If you use the folder tree or any other UI element outside the miller container to browse into a folder "far away" (e.g. typing a path in the addressbar), then the current miller pane is reset to the new folder. But miller mode remains active, so if you browse subfolders of this new master folder, sideways panels will pop up. To turn off miller mode completely, use **File > Column mode** menu or click on the equivalent toolbar/ribbon  button. Also note that miller panes will not survive a program restart, only the last active folder will be re-loaded when you start explorer<sup>2</sup> next time.

## Namespace extensions

Windows shell is an extensible environment. It supports custom extensions that enrich your desktop experience. xplorer<sup>2</sup> is tightly integrated with windows shell so any shell extension you install that works with windows explorer, should work for xplorer<sup>2</sup> too. There are many kinds of plugins available to download, the most important of which are:

- Namespace extensions that add new special folder types xplorer<sup>2</sup> will browse as folders, like 7zip archives, SFTP and so on.
- Extra commands in shell context menu, that work with selected files.
- Preview handlers that allow [native](#) preview pane to show rich previews of new file types
- Property handlers that add extra [columns](#) showing file properties in detailed view mode xplorer<sup>2</sup> supports both modern property handlers and old school column handlers
- Text extraction filters that allow searching for text contained in documents

You can see the full range of supported shell extensions and download links in the recommended [plugins](#) page. Most of them are free to use. When you install programs, they usually add several shell extensions for search and preview automatically. You only need to install shell plugins manually if you find that you are missing functionality for a certain file type.



Shell extension plugins are developed by third parties, unrelated to xplorer<sup>2</sup>. Plugins are known to be buggy and unreliable at times, and may end up crashing and taking down xplorer<sup>2</sup> too. If you encounter frequent problems with a shell extension please send a [bug report](#).

Another class of plugins that xplorer<sup>2</sup> understands are those of the popular **Total Commander**, another windows file manager. There are plugins for extended folder browsing (packer WCX), preview and thumbnails (WLX) and file attribute & text content (WDX) for file search. There are tons of such plugins available to download for free, so if you cannot find a windows standard shell extension for some exotic file type, you might be lucky with a total commander plugin. TC plugins must be installed with a [plugin manager](#) to be integrated with xplorer<sup>2</sup> — whereas standard shell extensions are discoverable immediately once you install them.



Both shell and TC plugins are in-process DLL files, which means they must be 64 bit if you want to use them for your 64 bit xplorer<sup>2</sup> installation. Unfortunately some old useful extensions may only be found in 32 bit trim, so they would only work for 32 bit xplorer<sup>2</sup> (see the [installation](#) notes). But then you would need to find 32 bit versions for all other extensions you use as well! Welcome to DLL hell.

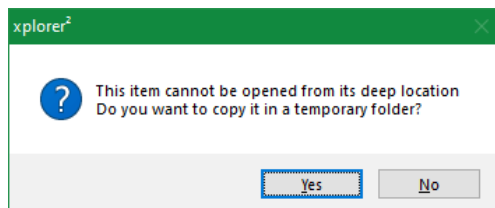
### Further reading

- Download a tool that reports on installed [shell extensions](#) for a file type
- Mount** VHD virtual box "hard disk" files as a drive letter and access their contents


### Deep path support

In the old era of 8.3 file and folder names, an overall path limit of 260 characters wasn't a problem — you could have folder hierarchies 27 levels deep and still fall within the `MAX_PATH` constraint. Ever since the 8.3 limitation was lifted and people started giving verbose names to their folders, the 260 character limit became a bottleneck. This concerns the overall path length so a few long folder names will add up and you'll hit the limit in hierarchies just 3-4 levels deep.


The low level NTFS filesystem does support long paths, up to 32767 characters long, and that should be enough even if you go crazy with long folder names — individual folders and file names are still limited to 260 letters. Super deep paths have a leading path descriptor that marks them as such (`\\?\E:\temp\deep_start\...`). Unfortunately at windows shell level things are still retrograde. xplorer<sup>2</sup> is deep path aware, and can happily copy and manage files beyond the 260 letter limit, but most other windows programs cannot, and this goes for most **programs** that open your files. Put a word document in a deep path and you will find that MS Word cannot open it — so what's the point using deep paths? The best xplorer<sup>2</sup> can do is to copy the deep file into a temporary location so you can at least read it (but you cannot edit it!)



Under the circumstances you must avoid long folder names so you keep your documents at levels above (less than) the 260 character limit. If you insist on using deep paths, you can provide shell access using folder [junctions](#), where you create a shortcut to a deep folder and place it high up in the hierarchy so that it looks "shallow" to deep path unaware programs.


 xplorer<sup>2</sup> does its best to help you manage files in deep folders, but it's being let down by windows shell. The following features don't work in deep hierarchies:

- Drag-drop *into* deep folders — use [robust copy](#) instead
- Deleting to the recycle bin doesn't work — items are deleted permanently
- Most shell context menu commands are unavailable, especially those installed by 3rd party apps
- Opening documents down deep may fail, or even crash the associated application.
- Opening text files with the internal [viewer](#) is ok


 The stock column **Characters** can be used to see the total path length of filenames. When this value goes near 250 you have reached the point of no return! Try to trim parent folder names if you can. You can use this stock property to [search](#) for files that are stored in deep folders (*characters > 260*)

## File properties

Modern windows OS offers a rich set of file properties, in fact there are over 1000 to choose from. Not all of them make sense for managing files, but there are certainly many interesting ones — including a few extra added by xplorer<sup>2</sup> called [stock](#) [S] columns. Certain column handler shell extensions also add to the property system, accessible to xplorer<sup>2</sup> like native properties (marked with [X]). Properties are pervasive in xplorer<sup>2</sup>, you can see them as file details or use them to select, filter and search for files — with consistent, easy to master property interface.

 The **ultimate** edition also allows [system properties](#) to be used as columns. At present they are imported is using "System property list" registry setting — available through the [advanced options](#) editor.

There are familiar properties like file Name, Size and date modified, and more exotic ones like Owner and Comment. Common properties apply to all [file types](#), but there are some that apply only to particular types, as dimensions for photos, duration for music etc. Properties that aren't applicable for a file type remain empty.

 Most properties are for regular filesystem items; [special](#) folders have their own content-specific properties, e.g. ThisPC has a *Free space* property that you won't find in regular folders. Compressed archive and cell phone pseudo-folders have basic file information (size/dates) and can inter-operate with regular folders.

File properties are divided in 3 main categories that determine how they are compared:

- **Text** properties like Name and Comment
- **Number** properties like Size and Duration
- **Date** properties like Date modified and Date taken for photos.

## Date formatting


Wherever xplorer<sup>2</sup> shows file date information, it uses your system default short date formatting, which includes date and time in a compact form like 06/11/2023 18:25:18. This default format comes from your system date options, which are determined by your region (locale) — using control panel or the newer Settings app.


Most people don't need millisecond time information for files and folders, and xplorer<sup>2</sup> can show user **friendly** dates instead of bland numbers. Sorting by date uses the **precise** file dates, but for display purposes something more palatable is shown, like *yesterday* or *last month*.

Name	Modified <sup>1</sup>	Modified <sup>2</sup>	Modified <sup>3</sup>
x2help.htm	07/11/2023 08:12:32	Last hour	6 minutes ago
strovolos.mhtml	24/10/2023 07:27:51	Last month	14 days ago
W820NB.mhtml	31/05/2023 11:06:46	This year	31/05/2023

The figure above contrasts precise dates [1] with 2 different friendly date options available. Method [2] uses big chunks of time durations (the same date categories used in [grouping](#)), whereas extra-friendly **file age** [3] has better resolution, which is at least one-day accurate even for very old files. If you prefer friendly dates, tick "User friendly date..." [option](#); if you want file ages [3], you must also tick "Extra-friendly file dates..." [advanced option](#) — otherwise method [2] is used.

NOTE: the [details](#) pane always shows precise dates, even when friendly dates are enabled

 When you turn on friendly dates, files are sorted by their exact dates regardless. So if two files show date modified **yesterday**, the one showing first is the most recent of the 2 (assuming that you arrange by date modified, descending — most recent files first).

 xplorer<sup>2</sup> has its own date and time format strings available through [advanced options](#) ("Custom date format"), which are used for precise date mode [1] (not friendly). Tweaking these format strings allows you to fine tune the date presentation, e.g. you may prefer the month name (Nov) instead of its number (11) — use the MMM month format string for this switch. Date and time format strings are explained [here](#). If you want to see millisecond information, add .zzz to the time format string.

**Further reading**  
→ [Curiosities](#) surrounding file and folder dates

## Folder views

We have already seen in summary how you browse individual folder contents in the [active pane](#). There are many tweaks you can apply to a folder pane, to suit your style and the particular content. Do you want to see all file details? or you prefer to see more files and check only the details of only one item at a time? Does the folder contain documents or photos? You can choose pane style by **View> Pane style** submenu.

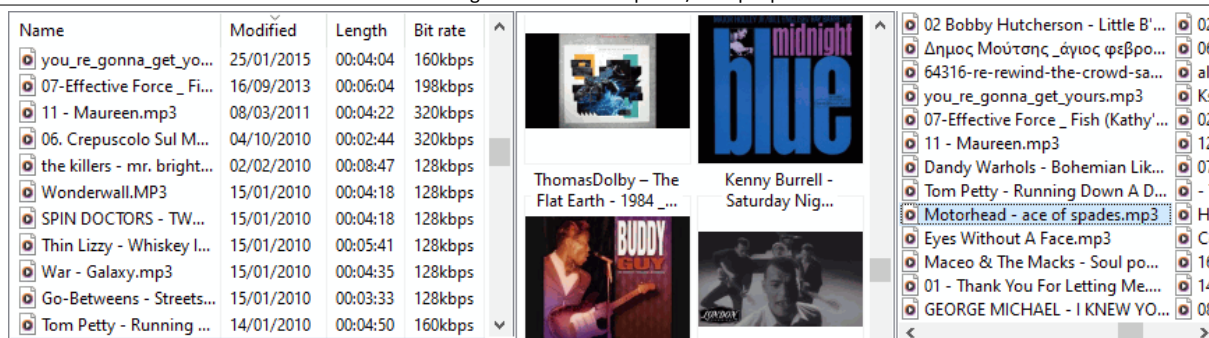


Figure 10. Details, thumbnails and list mode

Unlike windows explorer, xplorer² doesn't automatically change browsing modes depending on the folder content. It always uses the same pane style, the one you picked last. If you want to browse particular folders in a fixed style, use [custom folder settings](#).

## Sort options

When a folder contains many items, the **order** you see them becomes important. Do you want an alphabetic list of filenames, or you want to see the most recently modified item first? Any property can be used to sort files. The most basic sort modes are in **View > Arrange by** submenu (also available from a pane's background menu). In details view mode you can use the header row for easy sorting:



Click on the column header to arrange by the corresponding file detail, e.g. Name, Size, Comment or whatever else happens to show up as a file detail. Click on the same column again to **reverse** the sort direction from bigger to smaller and vice-versa. A little triangle shows the active sort mode (Name in the example above) and its ascending/descending direction. When you click on a date column, it defaults showing the *most recent* file first (descending), whereas text and numeric properties default to ascending order.

Header row is the only means to sort by "extended" properties that don't appear in **Arrange by** menu. If you tick the [option](#) "Show column header in all views..." the header will show even in list or thumbnail mode, facilitating easy sorting with the mouse.

A folder usually contains both files and (sub)folders. By default xplorer² groups the folders **separately**, but you are free to choose another bundling scheme, using "Folders sorted" program option. You can have them separately at the top or bottom of the list, or mixed with files — using whatever sort method is in force.

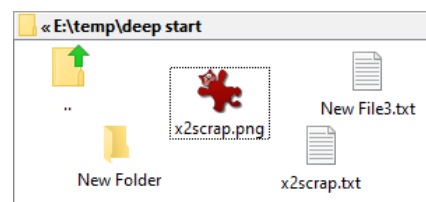
Text properties like **Name** are compared strictly alphabetically, which means that 10 comes *before* 2. If you want your files sorted numerically (e.g. file1, file2, ... file10), tick "Natural number sort" option in the Advanced property page of program options.

You can <Shift>+click on a column header to set it as **secondary** (or tertiary...) sort direction. This is most useful in [scrap](#) windows like file search results: you can have a primary sort by Path, then shift-click on Size column and have items that have the same path sorted by size.

## Manually arranged folders

In smallish folders set in large icon or thumbnail mode, items can be arranged as icons on your desktop. Use **View > Arrange by > Manually** menu command to move items with the mouse and leave them where you like. One possible use is to create "piles" of items to denote their mutual relationship, or "put aside" some files for further processing (but [color tags](#) may be more appropriate for this task).

The item positions can be saved as part of other custom [folder settings](#), so when you come back to this folder you'll find the items where you left them. Note you can sort (e.g. by Name or Date) a manually positioned folder if you need, temporarily destroying your custom positions, but once you switch back to manual arrangement, the positions will be restored.



There is another way to rearrange items **temporarily** in a regular list or detailed view. Select the item to move and press <CTRL+SHIFT+UP> (or down) arrow keys to move it. This can help in [mass renaming](#) when using position related automatic counters (to create names like n1, n2, n3...)

## Choosing details ▶ PLAY

In detailed view you can select the **columns** of information you want to see for your files. xplorer² supports all the standard shell columns and has its own extra columns too, called *stock* (denoted by [S] in figure 11). Use **View > Select columns** menu (or right click on the header) to pick the columns that are most appropriate for your task at hand.

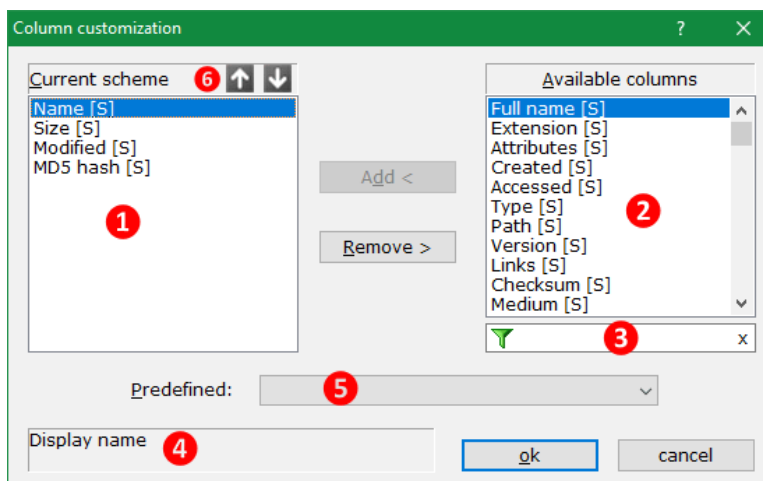


Figure 11. Column selection dialog

Current scheme box [1] lists the columns you have already chosen and show in the header. Click **Remove** button to remove the selected column from the current set. Columns

available for selection box [2] lists all available properties, included e[X]tended properties provided by shell extensions. Select a property from the available box and **double click** on it (or click **Add** button) to add it in the current column set.

As there are hundreds of available properties, use the filter box [3] to search for a column by name, e.g. type **date** to see all date-related properties that can be used. [S]tock properties have a description as well, showing extra information on what they do in box [4].

Use the arrow buttons [6] to rearrange column order in the header. You can also rearrange columns by drag-drop (when the column selection dialog doesn't show). Grab a column with the mouse and drop it where you want it in the header row.



Try to leave **Name** column first (leftmost). If you move it rightwards, then in-place file rename will behave funny (it will pop a small dialog window similar to **Change type** command)

You can save a group of columns as a set using **Customize > Column sets** submenu. Define a set of columns then use **Add** submenu to save the set with a descriptive name. The list of column sets is managed in a way similar to [bookmarks](#) list. You can have one column set for (say) photos, one for music and so on, and quickly switch among them using **Customize > Column sets** list. The **Predefined** drop-down list [5] in the dialog above gives access to such saved column sets for use in the dialog.

**Note:** The same column customization dialog is used to select columns for [infobars](#).



File detail columns are mostly fixed width. Once you resize a date column for example (using the header splitter bars between columns) you don't need to change it ever again as date format doesn't have variability. File names on the other hand vary a lot in length. To avoid manually resizing Name column each time you browse a new folder, tick "**Automatically resize name column...**" [option](#), which will try to fit the longest filename — up to a limit of 2 thirds of the available pane width. If any long name is clipped you can always rely on the tooltip to see the entire name.

## Show in groups view mode

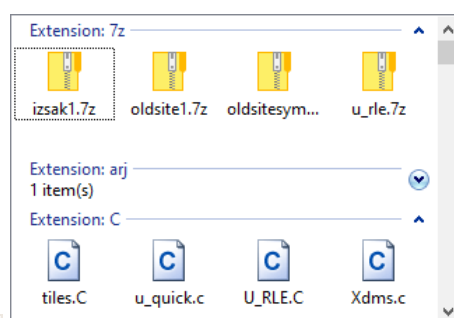
Grouped view mode is an accented sort method where items are bunched up into categories, based on the major sort property. Many file properties support arrangement in groups, but not all do. A property must divide files into chunks of similarity — depending on the **property type**: text, number or date. For example Names can be grouped by starting letter, file sizes as small/medium/large, and dates as today, this week and long ago. Properties like MD5 hash on the other hand, which are unique for each file, cannot be used as categorizers — each file would be in its own distinct category. Any files that don't support the categorizing property end up in *Unspecified* group.

To see a folder categorized in groups, sort by a group-compatible property, then use **View > Arrange by > Show in groups** menu. The groups will appear as in the pic to the right, where we are sorting by **Type** (file extension). Each category is defined by a particular extension, and all matching files are grouped together. Each group has a title e.g. *Extension: 7z*, followed by a divider line. At the right end of this line there's a little arrow you can use to **collapse** a group, hiding its items (as the *Extension: Arj* group in the example). Click on the arrow again to re-expand a collapsed group.

While showing groups, changing sort modes e.g. from Type to Date modified, will result in a different grouping appropriate for the new sort property. Normally items within each group are arranged alphabetically by name; to apply a different sort mode hold down <SHIFT> and click on the desired secondary sort property using the header row. To turn off groups altogether select **View > Arrange by > Show in groups** menu once more.



List view mode does not support grouping. The closest you can use that supports groups is **small icons** mode, but unfortunately this has many bugs and visual glitches for everyday use.



## Custom groups

Simple grouping is using the major sort property to break down files into fixed broad categories. Using custom groups feature you define your own categories, each with a membership rule. Use **View > Arrange by > Custom groups** menu to start defining your rules:

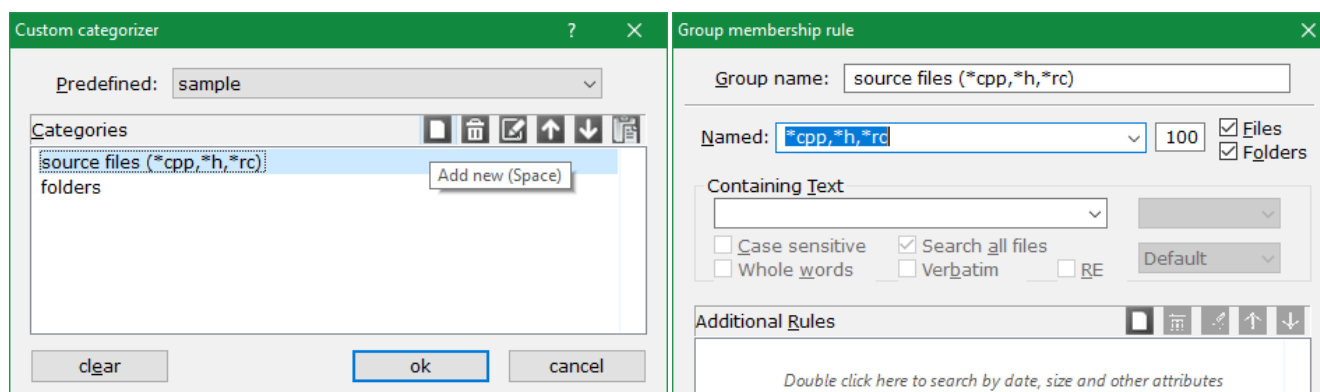


Figure 12. Defining custom groups

Using the small toolbar at the top-right corner (or the context menu) you can define, edit, reorder and delete custom categories using standard [hyperfilters](#) as they are used for file search. These may be as simple as a name wildcard, or a boolean combination of file properties. Each filter determines the group membership. Items are placed in the first matching category (in the order they appear in the dialog). Any items that cannot be categorized end up in the "unspecified" group.

The example in figure 12 will create (up to) 3 categories in each browsed folder, one will have files with CPP, H and RC extensions (it is wildcard based), the second will gather all folders, and all the remaining files will be in unspecified category.

Once you define a custom grouping you can save and reuse it using **Customize > Custom groups** menu. The group submenu list is managed in a similar fashion as [bookmarks](#) menu. Such saved groups are also available through the **Predefined** drop-down list in the above dialog.




If you want to change the definition of a custom group, you cannot use its list's **Organize** menu. Instead, first apply the old group to the active pane, then define a "new" group using **Arrange by > Custom groups**. You will see the old saved categories in the dialog above, where you can modify them, add/remove categories etc, then save them again with **Customize > Custom groups > Add current** menu with the *same* name — thus overwriting the old definition


If you want to stop arranging in groups (either simple or custom) uncheck the menu command **View > Arrange by > Show in groups** or click on the equivalent toolbar button.

## Folder view settings

xplorer<sup>2</sup> offers a sense of continuity as you browse your **filesystem** folders, maintaining your settings including view modes, sort orders, column widths, etc; it even remembers the item that was focused the last time you browsed a folder and brings it into view when (and if) you return.

In other words, xplorer<sup>2</sup> uses the current view settings for any pane as default. If for example you change from detailed view mode (**View > Pane style** menu) to list mode, then list mode will be default from then onwards for all folders browsed in that pane or tab.

- 

There may be more than one folder pane concurrently in xplorer<sup>2</sup>, browsing different folders. Each of these panes is an independent entity and has its own "defaults"; when you change one tab to details, or sort by Size, the remaining tabs will not change.
- 

When you browse into a [special](#) folder (e.g a zipfolder) and you happen to show file **details**, then columns will change. xplorer<sup>2</sup> maintains a *separate* default column set for each folder type: one for regular filesystem folders, one for zipfolders, one for ThisPC and so on. For better continuity when switching from filesystem to compressed folders, try using [stock](#) columns that are available everywhere.

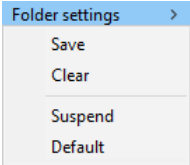
A different way to manage folder view settings is to setup a pane like you want it using various commands from **View** menu (e.g. pane style, number of columns, grouping, sort order) then declare this view mode as default using **Actions > Folder settings > Default** menu command. This way you tell xplorer<sup>2</sup> which default view mode to use for all folders. If you change any view parameters in some folder they will only last temporarily until you browse another folder, where the default settings will again return.

Note that folder settings can be different for each pane (left/right or up/down in dual pane mode), and even for each folder tab in a pane. You can have one pane in detailed view mode and the other in list mode. When you quit xplorer<sup>2</sup>, only the active tabs save their settings; next time you start xplorer<sup>2</sup> all tabs will look like the two last active tabs.


xplorer<sup>2</sup> does not change its view mode automatically to suit folder contents. You can force particular folders in a fixed view mode using **Actions > Folder settings** menu. So for instance you can have "My pictures" folder to appear always in thumbnails, even when you are browsing in a pane that is normally in detailed mode. When you get out of such customized folders the pane returns to its previous view mode. When you enter a folder with custom settings a [status bar](#) pane icon will appear as an indicator.


Under this concept, you setup a default pane style for most of your folders, e.g. details, and for the few exceptional folders you want to browse differently, save a custom view mode. When you first use **Save** menu to turn on a custom folder view, xplorer<sup>2</sup> asks you whether to apply the same mode to its **subfolders** — that may make sense if all subfolders have similar special content. Here is an easy way to do it:


- Enter the target special folder; its view mode will be the same as all other folders
- Use **Folder Settings > Default** menu to safeguard the default style
  - If you skip this step, you must change the pane style back to what it was once you come out of the special folder
- Change the pane style, sort order etc to customize the special folder
- Use **Folder Settings > Save** menu to fix the changes (for this folder only)



The menu command **Suspend** is an on/off toggle you can use to temporarily stop observing custom folder settings. When enabled it is as if there are no special folders.

- 

If you make view changes inside a folder with saved custom view settings, they are ignored — the folder is frozen to the original mode you saved, unless you use **Save** menu manually for the latest modifications. If you prefer xplorer<sup>2</sup> to update your custom folders automatically, tick the [advanced option](#) "Automatically save custom folder settings modifications"
- 

Try not to overuse the custom folder feature. It should be only used for a handful of exceptional folders. Don't use the feature to provide the default browsing mode in each and every folder you visit! At any rate, if you overdo the feature by mistake, use **Folder settings > Clear** menu to undo your customizations.
- 

xplorer<sup>2</sup> saves custom folder settings in the system file `DESKTOP.INI` which is used by the shell for various customizations. If hidden files are visible, you will see this `DESKTOP.INI` file created the moment you use Save command.

**Further reading**

 How to undo custom folder settings [en masse](#)



## Simple file management ▶ PLAY

If you worked with windows explorer — and who didn't? — then you will find that xplorer<sup>2</sup> can emulate its looks and functionality, no PhD required to do simple file operations using the mouse. This window snapshot shows a basic xplorer<sup>2</sup> window with most panes hidden, browsing a single folder, with a folder tree on the left and a preview pane on the right, just like windows explorer!

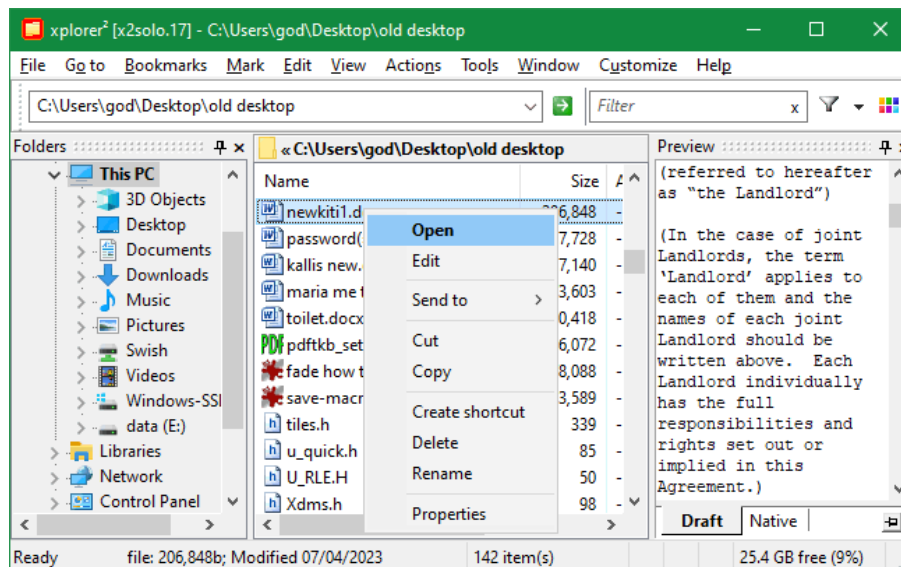


Figure 13. Windows explorer mode

Use the tree to locate folders, and select file items with the mouse. The preview pane will show a draft preview of the focused item. Double click to **open** the selected document in its registered application, or use the right click context menu to perform file operations as **copy**, **delete**, **rename** and so on (these commands are also available from **File** menu). To transfer copied files, switch to the destination folder and use **Edit > Paste** menu command, or use drag-drop. These are all elementary actions and will not be elaborated further.

The difference between **cut** and **copy** when we are dealing with files is that copy creates a *duplicate* copy where you paste, whereas cut-then-paste is equivalent to a *move* operation (relocation). Files that are cut appear somewhat dimmed.

If you hold down <SHIFT> key as you right click, you get an **extended** menu with more commands. For example, the context menu of executable files will include a **Run As** command that allows you to run a program impersonating a different user.

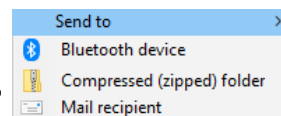
Context menu commands apply to all selected files. To build larger selections with the mouse, either use **lasso** selection (drag a rectangle around the files to be selected), or use keyboard modifiers as you click on items:

- To select a range of consecutive items, click on the first item, then hold down <SHIFT> and click on the last item.
- <CTRL> + click turns individual item selection on/off, without affecting any previous selection.

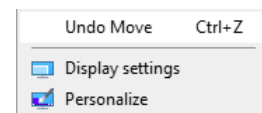
You must be careful selecting many items with the mouse because an accidental click may unselect all items. Use one of the advanced [selection methods](#) to build file selections with accuracy and safety.

An underused hidden gem is **Send to** submenu. You can create a compressed folder out of the selection, attach files to emails (**mail recipient** command) and more — you can even customize this menu for special targets.

Right click on some empty spot to open the active pane's [background menu](#). In there you will find commands to create **new files** and folders, change the pane style etc. Thus, just using the mouse and context menus, ignoring the entire menu system in xplorer<sup>2</sup>, you can do all the basic file operations!



All these basic file operations like drag/drop, copy/paste, rename and delete can be **undone**, albeit not by xplorer<sup>2</sup>, but from windows explorer! If you find you have made a mistake, **minimize** xplorer<sup>2</sup> and any other window that obscures the desktop, then right click on the background and you will find the undo menu command. As xplorer<sup>2</sup> uses windows explorer for its basic file operations, it is possible to undo them in this roundabout way.



### Further reading

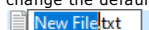
- Undo actions for [most dangerous](#) file operations

## Rename and change type

If you don't like the name of a file or folder, use **File > Rename** menu to change it. Although you can give long descriptive names to files and folders, it is recommended you kept names succinct — they take less space on the screen. Short names are imperative for folders if you want to create deep hierarchies without running into [deep path](#) problems.

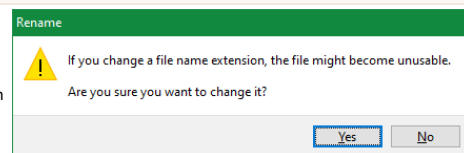
Almost all symbols can be used in filenames except for these that are used for path separators, wildcards and other reserved functions:  
 \ / : \* ? " < > |  
 If you use any of these in the name, they will be automatically replaced by underscores \_

When you create a new file or folder using **Actions** (or a pane's background menu), the item is created in the active folder and immediately you get in rename mode, to change the default New File.txt. We saw that changing extensions is potentially [dangerous](#) that's why xplorer<sup>2</sup> selects only the base name when you rename



Some people use <F2> as a way to **copy** the entire selected filename with <CTRL+C>, then press <ESC> to cancel the fake rename. Hence they don't like the partial name selection that xplorer<sup>2</sup> meant as "optimization". If you copy names more frequently than rename them, tick the [advanced option](#) "F2 rename selects whole name". Note if you are in list view mode (anything but details), <CTRL+P> copies the entire filename, so you don't need to resort to fake renames.

For plain text files, changing extensions is legitimate, e.g. changing a newly created file from New File.TXT to Stdafx.CPP. If you change the extension with inplace rename, you will be pestered by warnings against it. For fewer warnings of this kind use **File > Change type** menu command. This is also useful when extensions for known file types are hidden from system folder options.



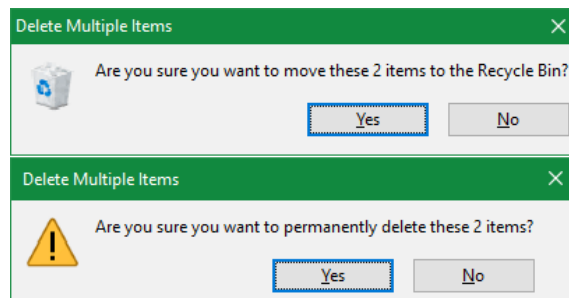
## Recycle and delete

File deletion is an inherently dangerous file operation, that can lose precious data by mistake or accident (always create [backups](#) of important documents). When you hit <DEL> key, the selected files and folders may end up in the recycle bin (which is recoverable), or **permanently** deleted, which is hard to undo — but [possible](#) if you act fast.

Only local filesystem files and those in USB-connected hard disks are moved to the recycle bin; other locations are deleted **directly**. Files that **cannot** be recycled include:

- Network files
- Files in USB sticks (flash drives)
- Files in phones and tablets
- Deep paths (length >256 characters)
- All non-filesystem items (e.g. zipfolders, FTP)

You must be extra careful when you delete files in any of the above folder types. It is recommended that you have delete confirmations **enabled** (using recycle bin [properties](#)) as an insurance policy. Delete confirmations that go into the recycle bin look slightly different than permanent deletions. Please study the following confirmations; the second one (permanent) requires special care because it is very hard to undo.



On the other hand sometimes you may want to **ensure** that files are deleted for good instead of being recycled. To delete permanently bypassing the recycle bin, press <SHIFT+DEL> keys together. If you prefer using the mouse, [customize](#) the toolbar and add the special **Trash** button . There's an even more extreme trashing option via **Actions > Shred** menu, which overwrites the file with random data then deletes it, making its retrieval impossible — but shredding isn't recommended for modern solid state (SSD) hard disks.

You can browse recycle bin in xplorer<sup>2</sup>, just like any other special folder in the shell namespace. Note that instead of the regular filesystem columns, you see details like *Original location* and *Date deleted*, which is information related to this special folder. Sort by **Date deleted** column to see the most recently deleted items first. Then select one or more deleted items and use **Restore** context menu command to undo their deletion:

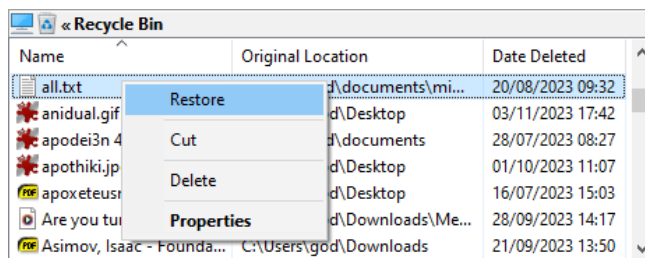


Figure 13a. Undoing file deletion

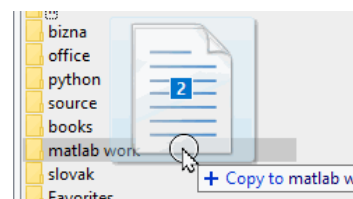
Don't forget to **empty** your recycle bin from time to time, using the context menu on the recycle bin desktop icon. If you accumulate too many recycled items, deletion will become sluggish.

## Super-charged drag-drop

Transferring files with drag-drop is natural and easy to grasp. Select a number of files, then grab them with the left (or right) mouse button and drop them to a destination folder. While dragging you get visual cues about what's being dragged and where will it go:

- A drag **image** shows the item(s) icons being dragged and their number.
- Eligible **drop targets** that will accept the dragged items are highlighted as you move around the pane; otherwise the folder pane itself will receive the drop.
- The impending file operation (Move, Copy or Link) is shown as a **drop tip**, that also includes the name of the target folder. If dropping isn't allowed you will see a no-go feedback cursor.

When you release the mouse button, you perform the last shown drop action into the last target folder. To abandon the operation without dropping, press <ESC> key or the other (usually *right*) mouse button, and the drag will be **cancelled**.



Dragged icons look nice but could get in the way of the drop target. If you don't want to see a dragged image, untick "Show images during drag drop..." [advanced option](#). This will also show simpler drop cues (a cursor for copy, otherwise it's going to be a move operation)

When you drag, you must find the desired drop destination. Most UI elements in xplorer<sup>2</sup> window accept drops, like the active and inactive folder pane, the folder tree and so on. If a pane contains too many items and you cannot see your target, you can **scroll** the window (while dragging) if you place your mouse pointer near the window edge where you want to move towards. You can drop to a different **folder tab** if you leave your mouse a second or two over a hidden tab — it will then activate to accept your drop. This *spring-loaded* folder feature also works in the tree, where you can open tree branches by hovering over them.

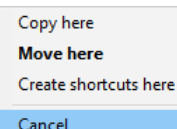
Drop targets are usually folders but you can also drop on program icons (EXE files, they will receive the dragged files as arguments to open them) — this method also works for [user commands](#) you place on a toolbar. Naturally you can drag-drop to other windows programs, not just inside xplorer<sup>2</sup>. Use <ALT+TAB> or the windows taskbar to select a different window to drop into.


If you find that you cannot drop *into* xplorer<sup>2</sup> from another program, you are most probably running elevated as administrator. The solution is to run xplorer<sup>2</sup> normally. If you need to drop into a protected system folder like Program Files, drag-drop will automatically elevate xplorer<sup>2</sup> momentarily anyway.

The default file operation when dropping is **Move** if you are working within the same drive (e.g. reorganizing files under C:\) or **Copy** when you drop to a different drive (e.g. from C:\ to E:\). You can override these defaults using keyboard modifiers:

- <SHIFT> to move files
- <CTRL> to copy files
- <CTRL+SHIFT> to create shortcuts

You will see the changing drop cues (label and mouse cursor) when you press these modifiers while dragging. If you don't remember which key to use, drag files with the **right** mouse button; when you drop, a context menu will open to select the desired operation!



 Drag and drop is easy and straightforward, but for some people is a plain *drag*. One tries to double-click and ends up doing an unwanted drag-drop (or vice versa) — especially with poorly setup mouse settings. An [advanced setting](#) "Turn off left drag-n-drop" will only allow dragging with the *right* mouse button, which should be easier to master and spare you the accidents. Use **Go to > Drop target** menu to browse the folder where you *last* dropped something inside, to recover accidentally dropped items.

#### Further reading

- More tricks with [drag-drop](#)
- Drag-drop to other windows and programs [demo video](#) 

## Previewing and launching items

Except for regular information like file sizes and dates, xplorer<sup>2</sup> offers access to the **contents** of files, as long as they are in normal or near-normal (e.g. zip) folders.

The most convenient option is to turn the *quick viewer* pane on from **View** menu. Then as you move the cursor around in a folder pane you can see the preview of the focused item. The quick viewer can show text (including RTF, Unicode and UTF-8), graphics, HTML, office documents, even audio and video files. There are 2 modes selectable by **tabs** at the base of the previewer panel. The figure below shows draft and native previews of the same document. The left one is plain text, the right one shows also formatting.

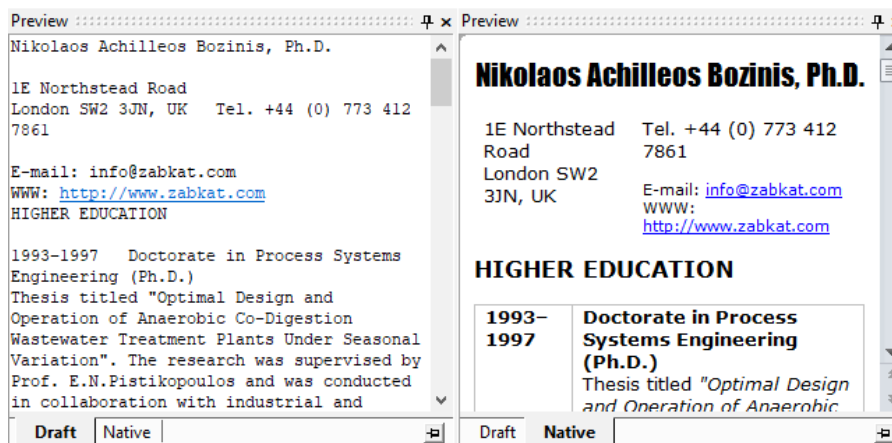



Figure 14. Draft and Native preview tabs

The **Draft** tab shows fast previews of most file types. For photos and graphics of all sorts you see a large thumbnail, which can be zoomed in/out and rotated through a toolbar  or touch gestures. xplorer<sup>2</sup> uses windows system thumbnail cache to offer quick image previews. It embeds Windows Media Player to offer previews of music and movies.

For plain **text files** like TXT or CPP, it loads and displays the first few KB (the limit is [adjustable](#)). Using the context menu (see right pic) you can change the preview font (**Set Font** command), select and copy text, **Search** for text in the preview etc. Normally xplorer<sup>2</sup> can figure out the encoding automatically; for the rare cases it won't, use **Encoding** submenu to pick a code page or indicate the text format (UTF-8, unicode etc). If you turn **Tail mode** menu command on, xplorer<sup>2</sup> loads the *last* few lines of the text file instead of the beginning. This makes sense for log files. Finally if you turn on **Text only** menu, xplorer<sup>2</sup> will show all file types as text — e.g. images will show a hex dump (47 49 46 38 GIF8 00000) instead of the thumbnail. Select this menu item again to turn on regular image previews.

Advanced office document types like PDF, DOC and XLS are also previewed as *unformatted* text using text filters meant for desktop search. When a document was scanned for text (e.g. a file [search result](#)), any matching keywords are **highlighted** so you can quickly see the surrounding text context. Use **Find next** menu to find the next search keyword match.




The internal text [viewer](#) called **editor<sup>2</sup>** is also aware of search hits. If you open a plain text file (a search result) in editor<sup>2</sup> using **File > View** menu, then <F3> key will find the keyword you searched for in xplorer<sup>2</sup>!

Copy	Ctrl+C
Select All	Ctrl+A
Search...	Ctrl+F
Find Next	F3
<input checked="" type="checkbox"/> Word wrap	F2
Set Font...	
Encoding	>
Tail mode	
Text only	

The little pin at the right end of the previewer's tab strip is the equivalent of **View > Lock viewer** menu command. When you lock the preview panel, it will stop following the focused file and remain fixed to the present file. A possible application is to listen to music while you browse. Unpin the previewer to resume normal previewing as you click around the panes.

Click on **Native** tab to switch to more accurate, slower previews. This mode is identical to the preview panel in windows explorer. To see accurate previews for your file types, you need to install preview handlers. Luckily when you install an office suite, it automatically adds preview handlers and text filters for its associated file types. If you have problem with certain document types, you can [search](#) for preview plugins.

As you can see in figure 14 above, native preview looks nicer, and you can preview the **entire** document, not just the first few lines. However it feels cramped in the narrow space available, so opening the document in its native program may be the better option. Draft mode is quicker and adequate for most preview needs.

 Some preview handlers are buggy and end up **locking** files you preview in native mode. xplorer<sup>2</sup> tries its best to release documents you want to delete, but sometimes this isn't possible. In worst case you can quit xplorer<sup>2</sup>, delete the locked document, then restart the program.

#### Further reading

- If you have problems previewing or searching for text in PDF documents, install [Sumatra PDF](#), a lightweight PDF reader that works very well with xplorer<sup>2</sup>. If you have another preferred PDF program, sumatra will let it be and install itself as a secondary handler. Just make sure you tick its **shell extension** options (for desktop search and windows previews)

## Peek preview

The previewer is one of the [adjustable](#) panes so you can resize and dock it to various locations. To see bigger previews you can use the splitter bar to resize the pane. But this larger preview comes at the expense of other panels, that end up shrunk.

Instead of a permanently docked preview pane, an alternative method is **View > Peek preview** menu, which pops a large window previewing the selected item. Hit any key (except PageUp/Down) and the preview goes away! The preview is similar to the **Draft** quick preview tab, so it will show images and plain text (including text extracted from DOC/PDF/XLS documents) only. For music/video or accurate **Native** preview you must still use the regular quick preview panel.

Likewise you can press both left and right mouse buttons together **simultaneously** to preview the file under the mouse cursor. Just enable "Mouse peek preview" from [options](#). Then let go of the mouse keys and the preview will go away!


A variation of this command can show popup previews for **two** files simultaneously. Select 2 items that have thumbnail previews (e.g. photos) and press <Alt+Shift+Q> keys and a popup window will show their large previews side by side for easy comparison. This is especially useful when considering [similar photos](#) before you decide which one to keep. Press any key or click the mouse to hide the preview window.

NOTE: if just one item is selected and you are in dual pane mode, this command compares the focused items in the left and right folders.

## Starting programs

You can launch documents using their associated program, by hitting <Return> key, double-clicking or right-clicking and picking **Open** from the shell context menu. If you need to start a **program** directly, there are many ways to do it with xplorer<sup>2</sup>:

- Double click on the executable file if you can see it in the active folder
- Use the [mini scrap](#) pane to store frequently used applications
- Define a [user command](#) then add it to a toolbar for easy access
- Type the program name in the [addressbar](#) if it can be found on system %PATH%

Nowadays the easier way to start programs is through the system **Start** menu. Press window  key and start typing your program name, and it will be found automagically! Only use one of the methods described above if you need to pass arguments to programs.

## Extract text information

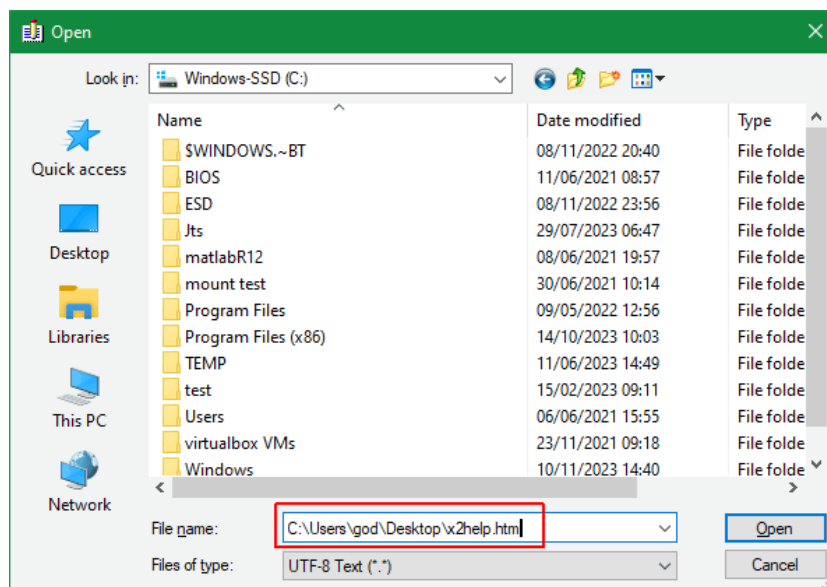
The contents of the active folder pane can be exported, as you see them in the window, with whatever file properties (columns) you have selected — even group headers, for e.g. *printing*. Use **Edit > Export CSV** menu command and your pane contents will be transferred to MS Excel (or whatever tool you have for CSV documents) for further processing. This command exports all the contents, regardless if they are selected or not.


**Edit > Copy columns** menu likewise copies all details, but only for **selected** files. The information is placed in the *clipboard* as text, which you can **paste** wherever you like. This command also works in the folder [tree](#) where it copies all expanded branches under the selected one.

If you are interested in filenames only, there are 2 options in **Edit** menu:

- **Copy path names** copies the full path of the selected item; if there are many items selected their paths are copied in *separate* lines. Holding down <SHIFT> would copy the space-free 8.3 path name, if available
- **Copy filenames** command omits the path and just copies the local filenames as clipboard text.


One good use of copying full paths is for **Open dialogs** used to open documents in most programs. These system dialogs are not as easy to navigate as xplorer<sup>2</sup>, so the alternative would be to find the file you want in xplorer<sup>2</sup>, press <ALT+C> to copy its path, then paste it in **File name** input box in open/save dialogs:




You don't have to find the exact folder to open, pasting the full path will do the job nicely! Here is a [demo video](#) 

On a different note, **Edit > Extract text** menu command is meant for formatted text documents like DOCX and PDF, where it will extract *pure* (unformatted) text and save it in a separate file, using text filter modules available for desktop search. For example text from ESTATEMENTS.PDF will be saved in ESTATEMENTS.PDF.TXT.

One possible use is to **compare** PDF documents for modifications, using a text comparison tool like WinDiff.

If there's no text to extract you will see an error message on the statusbar:  0 text files created

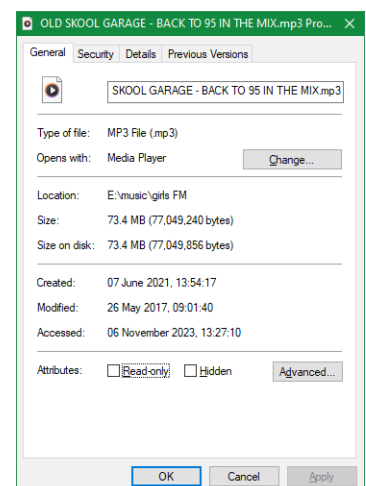
 By default extracted text is saved as plain text. If you are dealing with english or the text is in your default locale (main language setting) you should be ok. If on the other hand you want to save text as unicode, tick the [advanced option](#) "Paste text in folder saved as unicode". This will create bigger text files.

## Changing file properties



The filesystem shell supports many file [properties](#), simple stuff like size and file attributes, as well as more advanced ID3 tags, document information like subject and author and so on. Some of these properties are **innate** and cannot be changed directly (like file size in bytes), whereas others are adjustable like various file dates. You can browse file details in a number of ways:

- Using detailed view mode, where you can [choose](#) which properties to show
- The [status bar](#) and infobars show simple file attributes
- Use **File > Properties** menu to see the standard property page (see right pic). This shows properties for one or more selected files, organized in tabs. You can change properties using the **Details** property page tab.
- Use **View > Details pane** to turn on context specific attributes for each file type (see below).


xplorer<sup>2</sup> has many dedicated commands to change file attributes and properties *en masse* (french for many at a time :), like [dates](#), tags and [comments](#), [mass rename](#) etc. In this section we concentrate on simple property methods.



The most intuitive way to change properties is straight into the detailed view mode. Like in-place file rename, you can slow double-click (click twice but not too fast) on any property that is changeable, and you can edit its contents as such:

Name	Size	Modified	Keywords
 newkitil.doc.txt	3,536	28/12/2022 10:46:27	
 newkiti1.doc	206,848	07/04/2023	test tag;allo

You can think of the active folder pane as a little MS Excel **spreadsheet**, where you can change **cells** of various properties, like date modified as above. All properties are edited as text, so you must make sure you stick to the expected format — or an error will result. In the above pic, Name, Modified and Keywords can all be modified in-place, but Size cannot (it is innate.)



Spreadsheet mode is accessible through the **keyboard** as well. Press <F2> key as if you were to rename the selected file, then press <TAB> a few times till you reach the property column you want to change. Any innate properties will be skipped. Press <ESC> key to cancel any unwanted changes to a property.

Details pane

Turn on this pane from **View > Details pane** menu to see general as well as file type-specific attributes. This [dockable](#) and resizable pane shows the attributes of the **focused** file in the active pane, and updates as you browse different files and folders.

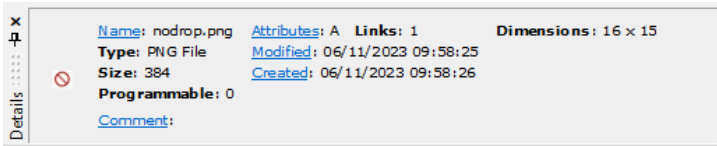



Figure 15. Details pane showing image properties

In figure 15 a PNG (image) file happens to be selected. We can see a little **thumbnail** preview on the left, followed by standard information as Name, size and date modified, but as this is an image we also see its **dimensions** in pixels. The most important type-specific information is shown e.g.:

- Dimensions for photos
- Duration for music and video
- Author and subject for documents
- Version and digital signature for programs



If you are familiar with HTML editing, you can change the file that defines which properties show in the details pane, or remove those you don't need. Any supported property can be added, both stock and shell. The procedure is explained in this [blog](#). Details are extracted lazily in a background thread so you don't need to worry adding slow properties here.

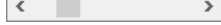
Properties that appear as [links](#) (like name and comment in figure 15) can be **modified**. If you click on such a property link, it's like you used the menu command that changes said property. Innate properties like **Size** just show a bold title, since they cannot be changed directly.



## Industrial scale folder management

If your folders contain thousands of files each, or you manage files from [multiple](#) folders simultaneously, you need heavy-duty tools for fast and reliable operations. xplorer<sup>2</sup> helps you move around efficiently within large folders, focus to a subset of your files with **filtering**, and finally **select** (mark) items for further processing.


In large folders, most files are hidden from view, especially if you are showing full details (one file per line). A standard horizontal or vertical scroll bar

 will appear, giving you an idea how many items are hidden and about your current location within the folder. You can operate your scroller with the mouse to see hidden items, but for more precise control use the keyboard to roam the active folder pane:

- Use the up/down or left/right arrow keys to move one item at a time
- For longer jumps press <PageUp> <PageDown> to scroll a "page" at a time, or <Home> <End> to reach the first and last items in the pane
- Typing the few first characters of the file you are after (incremental search) — e.g. type **AB** to jump to the first file that begins with these letters, say **ABSOLUTE.TXT**. If you type more than 4 letters, you activate the [quick filter](#) box
- Holding down <SHIFT> as you type will incrementally match file extensions (locate file types) — e.g. type <SHIFT+T> to jump to the first **TXT** file
- <ALT+DOWN> (or UP arrow) will jump to next (or previous) selected item — useful when selection isn't consecutive

As you move around, the **focused** item is changed and usually cancels all previous selections too. To move the focus without affecting the existing selection hold down <Ctrl> or turn **Mark > Sticky selection** on. This will help you to hand-pick items scattered in the folder listing (similar to explorer's checkbox selection mode).



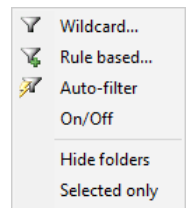
In large folders it may not be clear how many items are selected; you can see a few in front of you, but there could be more **hidden** (scrolled away) from view. Use the selection pane  on the [statusbar](#) (pane [\[4\]](#)) to double check how many items are selected (61 in this case) before you issue any file management command. If you don't see *any* of the selected items, use **Mark > Selection > Show** menu command to scroll them into view.

## Filtering for visibility


In large folders you can focus on a subset of items with filtering. A visual filter is a rule that matches one or more file properties; files that satisfy its conditions remain visible, otherwise they are hidden from view. Such filtered-out items cannot be seen or selected for management — just as if they don't exist any more!

There are several types of visual filters, activated from **View > Visual filter** submenu:

- Simple name **wildcards** like \*.TXT that allow only matching filenames.  
Instead of using the menu command you can also type a wildcard straight in the [addressbar](#) and press <ENTER> to apply it.
- **Rule based** (aka [hyperfilters](#)) are boolean expressions that combine file property conditions, and optionally contained text stipulations.
- **Auto-filter** command pops a menu that lets you select one of the file types present in the current folder.
- **Hide folders** is a simple filter that only lets files to show.
- **Selected only** is a manual and temporary filter that hides any items that are not selected, in case you want to concentrate on the selection only.
- The **On/Off** command turns the visual filter on and off (toggle). It is a quick way to reinstate the last used proper visual filter



Except for this visual filter submenu, you can use the [quick filter](#) box next to the addressbar to type part of a filename and show only matching items. There are even filters using [color tags](#), a topic discussed in the sequel.

Whenever any filter is hiding items from view, you will see a green funnel  icon on the statusbar. Double click on this icon (or use **View > Show all** menu) to cancel the filtering and see all the items in the folder.



When you turn visual filtering on/off, any prior selection is preserved. Say you had 10 items selected, and then apply some visual filter, any items (of these 10) that remain visible will be selected. Likewise selection is preserved when you switch pane styles or sort order.

## Wildcards for text matching

Wildcards are simple string matching templates using the special characters \* and ? as such:

- **\*** matches any character or substring
- **?** matches exactly one character

You can combine letters and these special characters to form a suitable wildcard for the matching task. They are most frequently used to filter filenames, but they are also good for any *text* property as we shall see later. They are commonly used to match file types (extensions), but you can use them for more specific text matching as these examples:

- \*.TXT would match 1.TXT, ABC.TXT and any file that has the TXT extension.
- ABC.\* would match ABC.TXT, ABC.DOC or any other file extension
- \*.\* would match any filename of any type, as long as it contains a dot — leaving out most folders!
- A?C.TXT would match ABC.TXT, A1C.TXT and all such variations of the second letter.
- A\*C.TXT would match ABC.TXT, A123C.TXT and all filenames that begin with A and end in C.

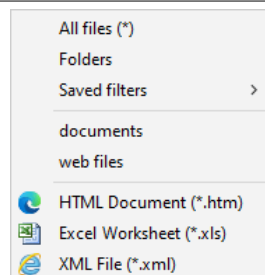
You can filter for **multiple** wildcards as well, if you separate them with **commas**. For example \*.TXT, \*.DOC would match any files with TXT **or** DOC extensions. A comma-separated wildcard expression is equivalent to **boolean OR** (any) operator. Use a *leading* minus sign to exclude a wildcard (boolean **NOT** operator), e.g. -\*.TXT will match any file as long as it is **not** TXT.

xplorer<sup>2</sup> can filter on [regular expressions](#) too, which are a more complex kind of text wildcard. Regular expressions are supported in [rule-based](#) filters.

## Automatic type filters


**View > Visual filter > Auto-filter** menu command is like wildcard filtering for **file type** (extension), which you don't need to type manually. When you choose this command (or click on its toolbar button) xplorer<sup>2</sup> examines what kind of files exist in the current folder (active pane) and pops a menu with them — and some more commands as you can see to the right. In this example, the folder contained a few HTML, some XLS and some XML files, which are listed alphabetically. Selecting one of these menu items will show only a particular file type, just as if you typed the equivalent wildcard (e.g. \*.HTM) for visual filtering.


Auto-filter menu also lets you show only **Folders** and cancel the visual filter (show **All files**). **Saved filters** is a submenu that lists all rule-based hyperfilters that you have [saved](#); select one of these saved complex filters to apply it on the current folder (it doesn't have to be wildcard based). Finally you can filter for broad document [categories](#) (search for "File types" in Global tab) defined in the registry as comma separated lists of file extensions. The default definition for "web files" for instance would show both HTM and XML files (but not XLS).



## Hidden files and global filters

A special kind of file property is **attributes**, a set of indicators that advertise the characteristics and status of each file (regardless of [type](#)). The default detail view contains a column for file attributes (usually shows ---A--- for the **Archive** attribute). Files marked as hidden or system are special, usually reserved for the operating system, and are not interesting for most users — who concentrate on their *own* documents.


When the program option "Show hidden files and folders" is not enabled, then any files or folders that are marked as **hidden** (have the **H** file attribute) are not shown at all. This is like an implicit filter on the file attribute property. The little eye  icon will appear on the statusbar when hidden files are obscured in this way.

 Even when show hidden files option is enabled in xplorer<sup>2</sup>, you may not be able to see some system files like THUMBS.DB if the system folder option "Hide protected operating system files" is ticked. To see all these super-hidden files, either untick the latter folder option, or tick "Show all hidden files, including protected" [advanced option](#) in xplorer<sup>2</sup>.

One way to block a particular file or folder from showing in xplorer<sup>2</sup> is to [change](#) its attributes to include H (assuming it doesn't already have the attribute). Then it will not be visible if you turn off *show hidden files* option.

xplorer<sup>2</sup> supports a global exclusion filter for both folder views and tree, hiding matching files and folders. The [advanced setting](#) "Global exclusion filter" is defined as a comma separated list of wildcards to **exclude** from view (alternatively it can be a [saved hyperfilter](#) name defined earlier). For example \*.DLL, \*.SYS would define a global filter that hides any DLL or SYS files permanently.


When you define a global exclusion filter, the hidden files statusbar icon will show as a reminder. You cannot turn off a global filter, it is always enforced. As it is pervasive, only relatively fast file properties are allowed. If you attempt to use a slow hyperfilter here, it will be ignored.

 Files and folders that are blocked by the global filter, or those that are blocked by their Hidden attribute, will not show up *anywhere* in xplorer<sup>2</sup>, including the folder tree. Hidden folders will not be **searched** either, unless you enable the hidden files option.



## Selection engine

Before you execute any menu command, you must choose the items you want to operate upon by **selecting** them. Sometimes this is easy, e.g. selecting **everything** by pressing <CTRL+A> keys, but in large folders you need the precise selection methods offered by xplorer<sup>2</sup> selection engine. Let's review the basics first:


- All the usual mouse gestures are supported, including <Shift>+click, lasso selection etc
- Middle mouse button (or scroller click) toggles the selection of the item under cursor — <CTRL>+click does the same job
- Using the keyboard, <SpaceBar> and <Insert> toggle the selection status of the focus; <Insert> key also moves the focus to the next item so you can select consecutive items
- <CTRL> + any arrow key moves the focus without affecting the selection

 Sometimes [sorting](#) items beforehand assists their selection. For example if you need to select all recently modified files, arrange by date, then all recent files will be consecutive on top and can be selected with <SHIFT+HOME> keys — starting from the last item.

These simple selection and unselection methods work with one item at a time. You build a large selection adding more items one by one. This manual effort can get tedious and unpractical if you require many items selected — not to mention that one accidental unprotected click can unselect everything you worked so hard to select in the first place! To protect against such mishaps use **Mark > Sticky selection** menu; when this mode is on, all mouse clicks work as if you had <CTRL> key pressed. To resume normal selection mode, select **Sticky selection** again to turn it off.

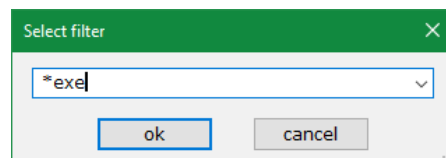
 Some people accidentally turn on sticky selection mode and wonder what kind of bug messed up their mouse. You can add the Sticky button (looks like a  flag) on the toolbar through [customization](#). Then you will have a visual indication if sticky mode is on (the button will be depressed). Click on the button to turn it off.


There are several commands in **Mark** menu that help you select and unselect multiple items in one go. Let's start with the simple **additive** commands, that select items on top of those already selected: (to start a fresh selection cycle, just hit <Alt+A> to unselect everything beforehand)

- **Select group** command marks filenames that match a [wildcard](#), or multiple comma-separated wildcards
- **Unselect group** likewise uses a wildcard to remove items from an existing selection
- **Select range** marks an exact number of items below the 
- Reuse a previously "saved" selection (see **Mark > Selection** submenu)

The above commands take a number or text input as a parameter. A simple text input dialog is shown for this purpose that hardly requires any explanation. You type your wildcard then click **ok** button to execute or cancel.

Using these elementary marking and unmarking mechanisms, either individually or combined, you can build collections of items whose complexity is only limited by the user's ingenuity!



 Sometimes it's easier to work in reverse, unselecting items — especially if you want most items marked except for a few excluded. In that case first do a **Mark > Select all**, then use mark menu commands that unselect items, or even manually <CTRL>+click on the items you want unselected. An equivalent strategy is to select items you want **excluded** first, then use **Mark > Invert selection** menu to select all the others!

Now let's discuss the non-additive marking mechanisms. These will *destroy* any previous selection, so they must be done first:

- **Total size** menu command can automatically select files (and subfolders) that tally approximately to a given byte size, so that you can optimize transfers to media with limited capacity, e.g. USB sticks.
- **Containing text** command will mark files in the current folder that [contain some text](#) in their data.
- **Matching a rule** is an advanced property [hyper-filter](#) to select and unselect items with ultimate precision
- Set the quick filter box to [selection mode](#) to select items that match a simple string — by filename or any other of the details shown in the active folder pane.

All these criteria-based selection commands affect *all* files and will *reset* any previous selection leaving only items that unambiguously pass the criterion in question. Typically such commands are performed first, possibly followed by elementary (un)marking commands to refine the selection. However when such criteria are used to *unselect* files (e.g. see **Unselect** button in [figure 16](#)) they only affect matching items.

**Mark** menu also contains a number of commands that affect the selection which are explained in other parts of this guide. Broadly speaking they consider all the files with a variety of criteria, and select files that match them. For example in dual pane mode you can mark items that need to be copied to [synchronize](#) two folders, or source files that must be [built](#) to make a compiled language project.



## Selection clipboard

You can save the current selection (filenames) using **Mark > Selection > Store** submenu. This can be used as a safe repository as you build complex selections step by step, especially with commands that erase the previous selection.

Once you have a selection stored, you can either restore it (**Select** command) or use the stored filenames to **Unselect**.

You can reuse a previously saved selection in the active pane as a *mask* via **Combine** menu. This advanced mode combines the existing and saved selection so that only items that are in both lists end up selected (boolean AND operation). So an item must be both already selected and part of the list stored with <Ctrl+F11> to remain selected; otherwise it is cleared. This command is handy for refining selection patterns.

Store  
Select  
Unselect  
Combine  
Show



You can reuse this special "clipboard" at a later stage to select (or unselect) items with the same names in a different *folder* or even in a different *window* managed by the same xplorer<sup>2</sup> process!

## Searching for text in files

xplorer<sup>2</sup> has a very good module to find text inside documents, both plain text files like TXT and C++ source code, and office documents like DOC and PDF — whose content can be read using special [text filter](#) plugins. One particular application of this feature is **Mark > Containing text** menu command, that lets you search for text inside files in the active pane, and select those that contain your search keywords. You specify what to search for and various options in the dialog and xplorer<sup>2</sup> will mark all the matching files.

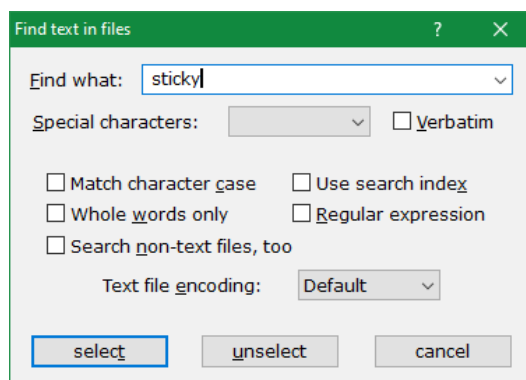


Figure 16. Find text in files dialog


The simplest use of this dialog is to type one word or phrase to search for in **Find what** input box, and click **select** button to start looking for text inside files that are shown in the active pane. Matching files are selected and any prior selection is cleared. If you hold <SHIFT> as you click on **select** button, any existing selection is preserved and matching files are *added* on top. It is also possible to **unselect** matching files using the respective button.



This command is quite slow, and some options make it even slower. You can help by using a [visual filter](#) to reduce the number of files scanned, narrowing the list down to those most likely to contain the desired text. xplorer<sup>2</sup> will enter an unresponsive state while a text search is ongoing — a progress bar on the [statusbar](#) will give you an indication of the time left. Pressing <ESC> key will abort the search.

This command finds text exactly as you type it, and internal spaces are significant (it's not like google search). For example searching for "this that" is different than "that this" (reversed order), and even "this that" (notice the extra space separating the words). Don't use quotes " unless you want to find a quote character. Think of it that you search for **phrases** instead of individual keywords. The advantage is that you can search for **symbols** as well as numbers and letters.

Several checkboxes in the dialog control how text is searched as below:

- **Verbatim**. This box turns off all special character processing, so you find exactly what you type. Tick it if you want to find commas and dollars inside files, without escaping them (see below).
- **Match case**. Tick this to force strict character case matching; if you type *This* it will not find *this* because the first letter doesn't match. Ticking this option makes the search faster; without it you get more search results.
- **Whole words**. Tick to enforce whole word matching. Without it, searching for *test* will match files that contain *fastest* as well as *testing*. Under whole words mode, it would only match *TEST* as a separate word.
- **Search non-text files**. By default only plain text files are searched (what you can open in Notepad), that are much quicker to handle. Tick this option to search in **all** types of files, including office documents (DOC, XLS, PDF, WPD, ODT etc) and even JPG pictures! xplorer<sup>2</sup> can extract text from office documents and search in them as if they were plain text (albeit much slower). Searching for text in pure binary files (that don't have desktop search filters) like JPG and MP4 is possible, but its utility is questionable.
- **Use search index**.  This option works only for [ultimate](#) edition. If you tick it, files will be found faster using windows search index. The down side is that you cannot find symbols this way. The keywords located can only be whole words or stems at the beginning of words — the usual limitations of google search.
- If you tick **Regular expression** box, the input string can be made of special metacharacters (see [Table 1](#) below). Regular expressions are complicated but can give you powerful text matching capabilities using wildcards, group constructs and more. For instance `abc[12]` would match both `abc1` and `abc2`. Regular expressions are much slower to process.  
Regular expressions imply **Verbatim** mode; all xplorer<sup>2</sup> special characters are ignored and only regexp syntax is used

You can search for special characters like tabs, newlines etc. In fact you can search for any character if you know its numerical equivalent, using the `$xx` token, where `xx` is the 2-digit hexadecimal value (e.g. `newline=10=$0A`). In the unlikely situation where you want to search for e.g. `$0A verbatim`, you must escape the dollar character and search for `$$$0A`. You will find some of these nonprintable characters in **special characters** drop down list.

The command automatically detects the **encoding** of plain text files (UTF8, unicode etc), if a Byte-Order-Mark (BOM) is present at the beginning of files. If BOMs are missing or you want to search within OEM files, you can force the encoding using the combo box supplied (see figure 16). To disable content interpretation altogether, including encoding

and text filters, pick **Raw** encoding; this will allow e.g. searching for BOMs or other binary content.

The quick [previewer](#) is aware of files that contain the text you just searched for. When a positive match file is [focused](#), the previewer will load the part of the file that contains the text in question, and will zoom around and [highlight](#) the hit, aiding further examination of contents.

### Boolean text searches

You can search for *multiple* text strings and also assign a boolean connotation to each substring. To achieve the former you just separate the expressions you are after with commas, e.g. **hello, world** will mark as positive hits files that contain either **hello** or **world** (or both). You can add a boolean effect using the special characters **+** for AND and **-** for NOT, at the *beginning* of each expression. E.g. searching for **help, +me, -god** will mark files that *must* contain "me", *maybe* contain "help" and *not* contain "god". (You shouldn't use spaces after the + or - characters, unless you want to match a string that starts with spaces.) The down-side of this flexibility is that if you want to search e.g. for commas verbatim, you have to either enter them as special codes (conveniently included in the **Special characters** drop-down box) or check the **Verbatim** checkbox to disallow multistring use. Also note you cannot combine boolean searches with regular expressions - you'll have to use the constructs in [Table 1](#) for equivalent boolean functionality.

**Table 1.** Special characters understood within regular expressions

Meta-character	Meaning
.	Matches any single character
[ ]	Indicates a character class. Matches any character inside the brackets (for example, [abc] matches "a", "b", and "c")
^	If this metacharacter occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets (for example, [^abc] matches all characters except "a", "b", and "c"). If ^ is at the beginning of the regular expression, it matches the beginning of the input (for example, ^[abc] will only match input that begins with "a", "b", or "c")
-	In a character class, indicates a range of characters (for example, [0-9] matches any of the digits "0" through "9")
?	Indicates that the preceding expression is optional: it matches once or not at all (for example, [0-9]? matches "2" and "12")
+	Indicates that the preceding expression matches one or more times (for example, [0-9]+ matches "1", "13", "666", and so on)
*	Indicates that the preceding expression matches zero or more times
??, +?, *?	Non-greedy versions of ?, +, and *. These match as little as possible, unlike the greedy versions which match as much as possible. Example: given the input <abc><def>, <.*?> matches "<abc>" while <.*> matches "<abc><def>"
( )	Grouping operator. Example: (\d+), *\d+ matches a list of numbers separated by commas (such as "1" or "1,23,456")
{ }	Indicates a match group
\	Escape character: interpret the next character literally (for example, [0-9]+ matches one or more digits, but [0-9]+\ matches a digit followed by a plus character). Also used for abbreviations (such as \a for any alphanumeric character). If \ is followed by a number <i>n</i> , it matches the <i>n</i> th match group (starting from 0). Example: <{.*?}>.*?</\0> matches "<head>Contents</head>"
\$	At the end of a regular expression, this character matches the end of the input. Example: [0-9]\$ matches a digit at the end of the input
	Alternation operator: separates two expressions, exactly one of which matches (for example, T the matches "The" or "the")
!	Negation operator: the expression following ! does not match the input. Example: a!b matches "a" not followed by "b"

For further information and regexp samples you can see this [reference](#)

## Defining hyper-filters

xplorer<sup>2</sup> can pattern-match filenames, search for text contained in files, and retrieve hundreds of file properties. These capabilities are brought together in **hyper-filters** which match files and folders using a combination of name, contained text and file property rules. A hyperfilter can be as simple as a name wildcard, or as complex as a boolean combination of many file properties and contained text stipulations simultaneously.

These complex filters are pervasive in xplorer<sup>2</sup> user interface, and a common dialog structure is used to define them, regardless if they are used for visual filtering, rule-based selecting or file search. Once you get the hang of them, you can use them everywhere to match files with precision. Here is an example dialog for visual filtering (**View > Visual filter > Rule based** menu command); if you compare it with [find files](#) dialog, you will see that are very similar.

**Figure 17.** Rule based filtering dialog


Hyper filter dialogs have 3 sections that define the filter (name [\[1-3\]](#), contained text [\[4\]](#) and additional rules [\[5\]](#)). At least one section needs to be filled in with your file matching parameters. Empty sections appear disabled (ghosted) to minimize visual clutter in this admittedly complicated dialog. Let's have a quick look at the input fields:

- ① **Named.** A [wildcard](#) to search for particular filenames or types (e.g. \*.txt, \*.dat). If you type a name without any special characters (\* or ?) it is assumed you search for a partial name (e.g. windiff is expanded to \*windiff\*). The drop-down portion includes some common document **categories** (e.g. {audio}, {pictures} etc) when you want to match broad file types.
- ② This label-less box turns on **fuzzy** name search mode. The number dictates the percent *exactness* when matching file names. 100% means exact match, but lower values will allow for small spelling errors or the presence (or lack) of accented characters e.g. umlauts and other diacriticals. Setting the fuzzy indicator to 0 turns on the **regular expression** mode where the Named field is interpreted as a [regular expression](#) to match e.g. abc[1-5].
- ③ Checkboxes for limiting the search to **Files** and/or **Folders** (or both).
- ④ Contained text criteria, when you are after files that must contain some text string. All the functionality of [find text in files](#) command is available embedded, including

all optional checkboxes (case sensitivity, whole words etc), special characters like newlines, encoding selection — the lot! You can search for a phrase or multiple comma separated keywords. Regular expressions can be used too, ticking the **RE** checkbox. When you search for text, it is implied that you don't want to match folders

- ⑤ **Additional rules** box lets you add any property as a file matching condition, e.g. find files up to a certain **Size**, modified before a certain **date** and so on.

You can define hyper-filters with as few or many criteria as you want. You can go from a nearly empty filter (e.g. all fields clear except for **Folders** box that will find all folders) all the way to any arbitrarily complex filter with tens of rules combined in boolean contexts. Only items that match *all* aspects of the filter are considered hits.

 The more rules in a filter, the longer it will take to process it for each file. Some properties (e.g. **checksum**) and contained text searches are the most time consuming. Still xplorer<sup>2</sup> checks easier rules first and leaves slow ones for last, optimizing searches as much as possible. Users needn't worry ordering the rules, this is done automatically.

When a filter dialog shows up, its fields are loaded with the filter you used *last time*. If you want to define a new filter, click **clear** button to empty all fields. Note the button controls in section [7] depend on the command, it's **filter** for visual filtering, **select** when you are going to use the filter for selection and so on — they act like the typical **OK** button, that closes the dialog and executes the filtering command.

## Saving and reusing filters

If there is a complex filter you use frequently, then it makes sense to save it and have it easily available. The individual input boxes remember their past inputs, but there's also a way to save the *entire* filter definition, including all parameters as a bunch.

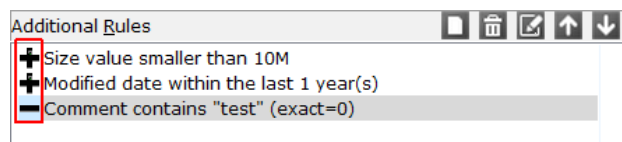
To store a filter, type a descriptive name for it in the **Predefined** [6] combo box and click on the **Save** button (see figure 17), and the current definition will be added in the drop down list. From then on you only need to select the saved filter from the Predefined list and its rules will be automatically restored for you.

To modify a previously saved filter, make your property rule changes and save it again with the *same* name — thus overwrite the old definition.

To delete a saved filter that's no longer needed, first select it, press **Clear** button to delete all its rules and finally click on **Save** button to confirm the deletion. Arguably this isn't very intuitive; the dialog UI is streamlined for saving and using complex filters, which is what you do most often anyway.

## Property rules

xplorer<sup>2</sup> has full access to hundreds of [file properties](#) that can be *all* used as criteria in hyper-filters. The additional rules box [5] lets you add one or more properties to fine tune your filtering, e.g. you can find files that are smaller than 10MB *and* modified last year:



You can add, edit, reorder and delete rules using the little **toolbar** on the top right of the Additional rules list box (similar to how we manage [bookmark](#) lists). If you right click in the rules box, a context menu pops up with the same commands as on the toolbar. All the usual keyboard and mouse clicks are also recognized within the list, e.g. <DEL> key will delete the selected rule, double-click will edit a property and so on. Each rule has a descriptive text and an icon reflecting its *boolean context* (+ for AND, - for NOT and | for OR).

Searchable file properties belong to three broad categories: dates, numbers and text. This determines how the properties are interpreted & searched: number and date rules are matched as *ranges* between a Low and High limit, and text properties are searched for *contained text* (e.g. we are searching whether the comment property includes a keyword).

Click on **Add new** toolbar button to add a new rule. A dialog opens to select a property and set the rule parameters as such:

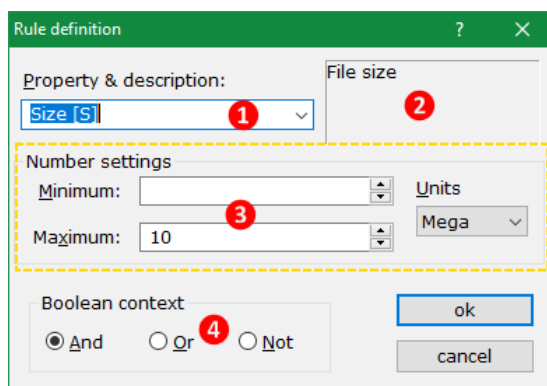


Figure 18. Sample numeric rule definition

First you select the property using combo box ①. There are hundreds of properties to choose from, but this box helps you by *autocompleting* the partial name you type against all matching property names. E.g. typing **DATE** will popup a window with possible completions, then you can select your desired date property (modified, created ...) from this [popup](#), either with the mouse or keyboard — not much typing required to find properties! A short description may show in ② if available (unfortunately not all properties have them). As you select properties of various types, the *middle part* ③ will change according to the current property.


Figure 18 shows the parameters for a numeric property (File size), which comprise a minimum and maximum limit. A file size *x* will **match** the numeric rule if it falls within the limits you supply ( $\text{min} < x < \text{max}$ ). Either limit may be missing if you want an open-ended rule; in the above example the minimum size is empty, so that makes the rule *Size less than 10MB* ( $x < 10\text{MB}$  for the mathematically inclined). If you are after an *exact* number, make both minimum and maximum limits **identical**.

Numeric properties support units of measurement using the **Units** drop down list. File sizes in bytes will be big numbers, so it is convenient to pick **Mega** as a unit — that implicitly multiplies your typed limit by a million (10MB  $\approx 10,000,000$  bytes). Properties with small values (e.g. picture width) don't need units, set them to 1.

**Minutes** unit is for matching audio and video **Length** (duration) property, measured in minutes

1
Kilo
Mega
Giga
minutes

### Further reading

• Demo video about MP3 information [units](#) 

Then comes the boolean context ④. When you define multiple property rules, do you want them *all* satisfied or *any* one match would be enough? All **And** rules *must* be satisfied, at least one of the **Or** rules and finally *none* of the **Not** rules. If you are not familiar with boolean algebra, just leave the default **And** radio button checked. If on the other hand you are familiar, you can have field day playing with hyper-filters!

Let's have an example. Say you want to find a file whose size is either smaller than 1K or above 2K but not inbetween. The simple solution would be to use two **Or** rules, one searching for files up to 1K and another one above 2K. Or you can use your boolean skills and define a *single* rule that says: **Not** "size between 1024 and 2048". Your boolean expertise saved you a couple of seconds there <g>



**Date settings**

☒ During the previous: 1 year(s)

☐ Between: ☒ 09/11/2023 & ☒ 09/11/2023

If you select a date property, the middle part ③ of the rule definition dialog will change as above, to accept your desired date range. Date properties can be defined as periods from the present time (**During the previous**) or as ranges (**Between**). Leaving any end of the range empty (untick the checkbox) removes the respective date boundary. For example if the low end of **Between** range is unticked, the rule will match any files that are *up* to the date specified in the upper limit box. You don't specify time (hour) limits directly. The low range is assumed from the beginning (00:00 midnight) and the high end includes the entire day (up to 23:59)


With **During the previous** mode you must select the time *unit* (years, months, seconds etc) and the quantity (number), so you can create rules like *modified during the previous 3 months*. This can also be expressed in **Between** mode with a low date limit 3 months from today, and the upper set to now — but it wouldn't be as clear to grasp.


**Text settings**

Find:

☐ Verbatim ☐ Regular expression


Finally, if you select a text property (like **Comments** or **Owner**), the middle part ③ of the rule definition dialog will change as above. You search text properties like you search for text in files: just type *part* of what you are after in **Find** box. All searches are case (and word boundary) insensitive for the broadest possible match. In this example if a file's Comment property contains any references of *test*, it will be a match, else no cigar. Wildcards (\* and ?) are also supported, unless you tick the **verbatim** option. Advanced string matching is possible ticking the **Regular expression** option.

 If you want to find all files that have *any* comment set, type a single ? (question mark) for the text to find. This wildcard requires one character to be satisfied, so it can detect the presence or not of any text property.

 Each text rule can contain many comma-separated substrings combined in boolean expressions as in [find text in files](#) command. For example, since the **Attribute** column is a string that contains letters A, H, S, R etc, searching for **+A, -H** within file attributes will match files that have the **Archive** but *not* the **Hidden** attribute (note the use of + & - to assign the boolean context for AND and NOT respectively). You can also use the **\$xx** notation to search for non-printable characters using the hexadecimal ascii code **xx**.

If you want to search for an **exact** text property, start your text rule with a colon : — it changes the nature of text match from *contains* to *is*. For example **:for** will only match the exact string *for* and not *form* or *proforma*. Leading colon also works with regular expressions.

After this long explanation, you managed to define a single rule; click **ok** button (see [figure 18](#)) to add it to the current hyperfilter — the new rule will show in **Additional rules** box. In the same fashion you can **Edit** an existing rule; double-click on it and the rule definition dialog will let you change the match conditions.


 Additional rules box lets you select more than one rule at a time using <CTRL>+click. If the selected rules are of the *same type* (e.g all text properties), you can edit them all in one go and set them to the same value or range. This feature may be used to search for a particular keyword in both (say) Comment and Tags file properties.

The up/down toolbar buttons in additional rules box are meant to reorganize the rules. When xplorer<sup>2</sup> is testing files against a hyperfilter, it processes the rules in the order they are **shown** in the rules box, so it makes sense to put fast rules first and slow rules towards the end of the list, to accelerate file matching. However xplorer<sup>2</sup> does this reordering automatically even if you won't!


## Easy filter and selection


Most of the work you do in xplorer<sup>2</sup> involves finding, filtering and selecting items, before doing something with them. The edit box next to the addressbar is a convenient way to do all these things from a single tool. Arguably it is one of the best productivity features in xplorer<sup>2</sup>!

By default the quick filter box is set to (visual) filtering mode. Instead of fumbling with [wildcards](#) and complex rules and dialog boxes to pass parameters, start typing what you want to find in the active pane and after the 4th letter the quick filter will automatically activate and show only files with matching names — you don't even have to press <ENTER> to start filtering, it is done **live** and as you type. If you type enough letters, the active pane list will shrink to a couple of items only, hopefully including the one you are after. To **reveal** all hidden files, click on x to the right of the box to clear the filter — **View > Show all** menu command does the same.

 To jump out of the quick folder box and return to the active folder pane, press <ESC> or <TAB> key.  
To jump to the addressbar press <SHIFT>+<TAB> keys together.

Normally the quick filter tries to match your input with **filenames** only; your partial filter string can appear even in the middle of filenames and it will be considered a match (e.g. type *hel* and you will match *help.txt*, *shell.dll* and any other file that contains *hel*. If you want to match text against all details shown in the pane (e.g. size, date modified and anything else that is visible), use the drop-down menu and tick **All properties** option. This type of filtering is *slower* therefore in large folders filter-as-you-type won't work — you need to press <ENTER> key to filter out files that don't match. You can even search for text inside documents ticking **Text contents** menu item, but that will make things even slower.

 xplorer<sup>2</sup> ultimate can search much faster in all properties and text contents if you tick **Windows search** menu. You can find whole words or stems (beginnings) quite nicely this way, if you browse folders that are in the search index. In professional edition this command is disabled.

Use the drop-down menu to switch to **Select** mode. The action button will change into . Instead of filtering now the box is used for *selecting* matching items. The idea is the same, what you type is matched against the filename or optionally all visible details (or even contained text). Press <ENTER> key to select matching items; those that don't match remain unselected — but visible.

Quick filter and selection mode work with the items in the active pane (it works even in the [mini-scrap](#)) — usually that means items in the current folder browsed. If you switch the box to **Search** mode from the drop-down menu, then it can search the current folder *and* its subfolders, like a regular [find files](#) command.

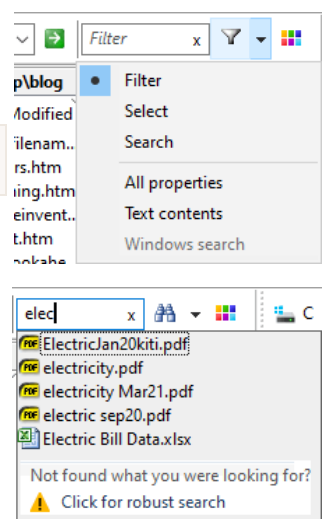


If you do a simple search for *filenames* only, matching filenames appear immediately in a popup window, and automatically update as you change the search keyword (live search as-you-type). You can preview and open files directly using the context menu on search results, or browse the containing folder for more file management actions. To jump into the search results pane press <TAB> key. As the popup is space limited, the search stops after 25 results are found; if you cannot see your target file, click the footer for **robust search**, which will do a regular full scale file search.

The live minisearch feature does not work if you tick any slow match options like **All properties**. In such a case a regular search is performed, in a separate window.

 You can adjust the behavior of the quick filter using [advanced options](#) — even turn it off altogether. Search for these options by name:

- "Incremental search autofilter" controls the number of keystrokes you type before the quick filter box steals the focus. Set this to 0 to disable this feature (you must click in the quick filter box to use it)
- "Filter box size %" sets the size of the filter box as a percentage relative to the addressbar size. Set this size to 0 to disable (hide) the quick search box altogether.
- If you don't like mini-search for filenames mode click "Disable quick filter box popup search"



The colorful button to the right of the filter box, is another way to quickly filter files based on their **color tag**. Please see the following section for more details.

Legacy quick search

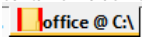
Before the introduction of the quick search box in the addressbar, xplorer<sup>2</sup> offered another type of quick search straight in the addressbar. This mode isn't as convenient because it doesn't update live the pane items as you type, but it is explained here for completeness. It's main objective is **selection** but it can also filter for matching items.


Despite their expressive power, fumbling with hyperfilters and text rules can be too time consuming. **Mark > Quick search** menu is a convenient shortcut that selects files that contain a phrase, either in their content or in some *text* column, including filename. It is a *local* search mode (scans items already in the pane) equivalent to a hyperfilter with one rule per text column, each matching your phrase in a boolean OR fashion (i.e. any one match will be enough); if no column matches the phrase, the command searches the text within the file. This latter slow stage can be skipped if not required, hitting <Esc>.

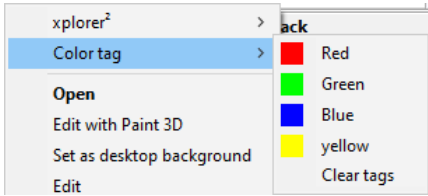
Quick search will only scan columns that are *visible* in the active pane. To make the most of this command you may have to activate a few interesting [columns](#), e.g. Comments, Version information etc, depending on your context.

Quick search uses the addressbar for text input. Instead of using **Mark** menu you can press <SHIFT+TAB> to jump to the addressbar and type your phrase, beginning with a colon, e.g. *:your phrase*, then press <Return> key to apply; any matching items in the active pane will be **selected**. If instead of marking you want to *filter* the matching items (hide everything that doesn't match), type your text and press <Shift+Return>. [Stored](#) hyperfilter names are also understood if enclosed with curly braces, e.g. `:{saved_filter_name}` is a valid input.

Manage color tags ▶ PLAY

Sometimes you need to mark a file for [later processing](#). One way is to place the item in the [mini scrap](#) pane. Another possibility is to color tag it, so whenever you browse its container folder it will stand out with a [color border](#) drawn around its name. Likewise you can color tag a folder in the [tree pane](#), making it easier to locate — then its tab's icon  will also be painted with the same color.


Color tags are set through the shell context menu (xplorer<sup>2</sup> adds a couple of its own commands at the top). Alternatively add the **Color tag**  button to the toolbar via customization.




- To tag a file (or a bunch of files), select them, click with the right mouse button, and from the **Color tag** submenu choose the color you like.
- To change the color of a previously tagged file, just apply another color using the menu
- To untag a file (cancel the color tag), use **Clear tags** menu command; the color border will be deleted and the file will show normally.

There are three predefined tag colors to choose from; you can add more colors with some [tweaking](#).

Adding a color tag won't change the actual files in any way (xplorer<sup>2</sup> uses a private database for color tagging information). You can use many colors for different kind of markings — the meaning of colors is entirely up to you. The color borders drawn around the names can be combined with rule-based [color coding](#) — but color tagging is much simpler and quicker.



Color tags are **permanent** and they are remembered even after you close and restart xplorer<sup>2</sup>. Color information is associated with a full **path** name. If you move a colored file to a different folder, thus changing its path, its color will be lost. Renaming a file preserves the color.

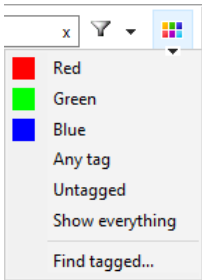


The color tag database is stored in the file %APPDATA%\zabkat\TAGDB.BIN. A quick way to erase all color tags is to quit xplorer<sup>2</sup> with <ALT+X> then delete this file — it is important that xplorer<sup>2</sup> is not running. When you restart xplorer<sup>2</sup> all color tags will be gone. Color tag borders won't show if you set item icons [setting](#) to **Plain**

Find color tagged files

Tagged items are hard to miss, their color borders are very conspicuous among normal unmarked items (you can make the borders even thicker using [advanced option](#) "Color tag border"). There are commands that **filter** a pane for its colored files; you can also **search** to find tagged files in subfolders.

You can quickly filter the active pane (*except* the folder tree) to show only items tagged with particular color, using the color filter button next to the [quickfilter box](#). This color button opens a submenu with all the filtering options. You can show only one particular color or all tagged items regardless of color (**Any tag** command). You can hide all color tagged items and only show those **Untagged**. In the end **Show everything** command will cancel any color filtering (you can also double-click on the [statusbar](#) visual filter icon).



This menu includes a **Find tagged** command that can show you color tagged files in other folders. In the tag search dialog first specify which colors you are after using the checkboxes, then state your search scope: do you want to search only below the current folder or everywhere?

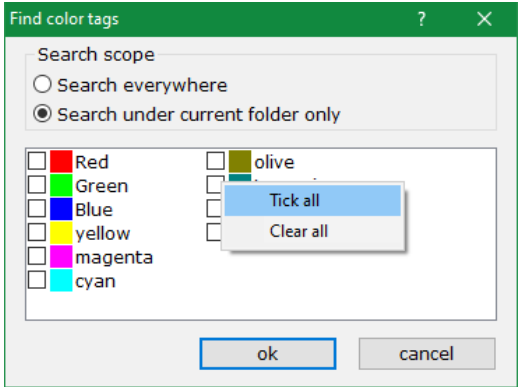


Figure 19. Find color tagged files dialog

Click **ok** button to start the search. If there are any tagged files with the colors you ticked, they will be presented in a standard [search results](#) window. If nothing can be found you'll get an error message instead — consider enlarging the search scope e.g. ticking **Search everywhere** option in the find tags dialog, or tick more color boxes (there is a **tick all** command in the right click menu).

The tag search dialog is very fast but will only search for colors. If you want to combine colors with regular file attributes like names etc, use [find files](#) dialog and add a rule on the **color tag** stock property. In this context tags are plain numbers in the order they appear in the color list (1=red, 2=green etc). This kind of combined search will be much slower though.

## Color coding filetypes ▶ PLAY

The various [file types](#) are shown with distinct icons so you can tell them apart. Items use normal font and text attributes, according to your desktop preferences (e.g. black text on white background). Windows explorer can show certain kind of files with alternate colors, making them stand out and easy to recognize, as for example *encrypted* files in blue. xplorer<sup>2</sup> extends this feature allowing you to customize the text color and weight for filetypes you are interested in. Use **Customize > color coding** menu to define or amend color coding rules:

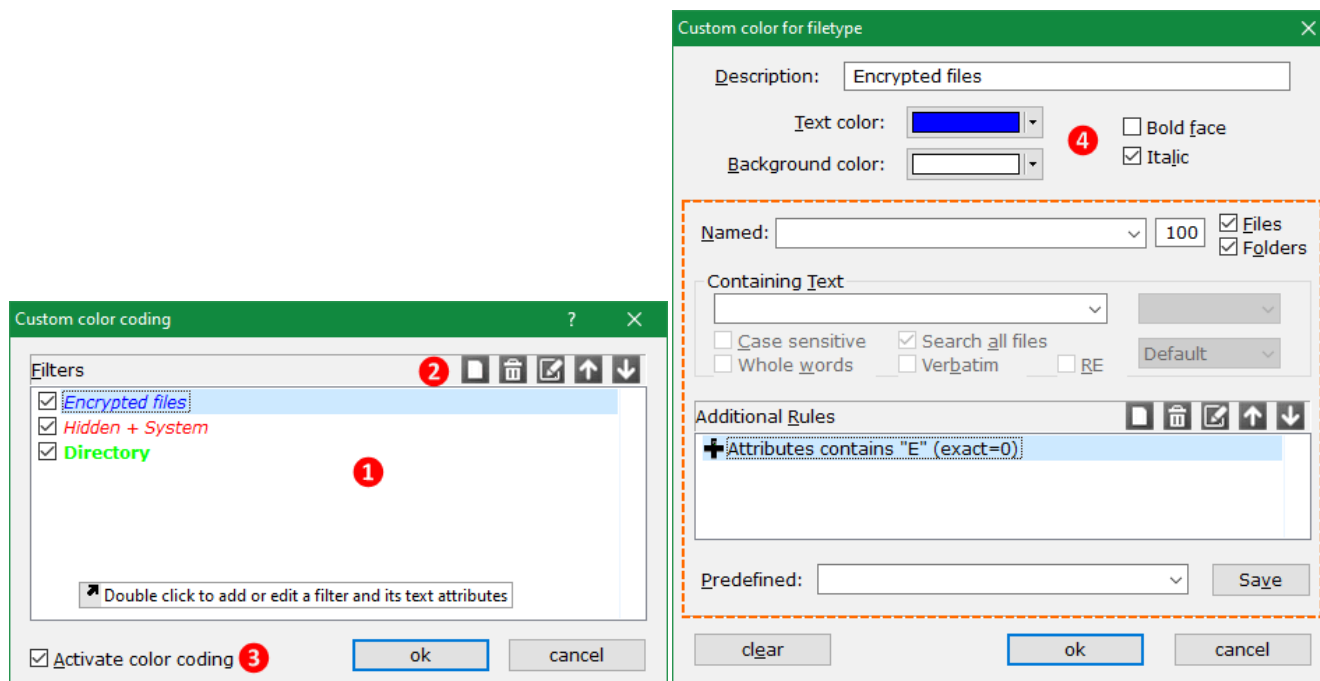


Figure 20. Custom color coding and definition dialogs

The custom color coding dialog (left) shows all the color rules you have defined in a list [1]. You can manipulate individual items (add/delete/edit) using the toolbar [2] and the equivalent context menu commands, as we saw [earlier](#). The list shows the exact effect (color and font) that each rule will have on matching items.

"Color coding filetypes" is rather a misnomer for what this system offers. Not only can you change colors, but also fonts (apply **bold** and *italic* effects). And we don't just filter on filetypes, you can use any file property using mighty [hyper-filters](#) to define a color category. For example the encrypted files category above is defined using the file attribute property (see dialog to the right) — of course you can also use a wildcard like \*.cpp to target a file type (extension).

Double click to add or change a color rule. The top section [4] is where you define the **text attributes**: foreground and background colors, and bold/italic style. The **Description** field isn't very important in terms of function, only as a reminder for your records — typically a summary of the filter. The biggest portion of the dialog is for defining the **membership** rule (compare with [figure 17](#)) — don't be put off by the apparent complexity. For simple wildcard color coding you would only use the **Named** edit control.

To turn on color highlighting tick **Activate color coding** [3]; clear the checkbox to return to normal colors. You can also turn individual color rules on/off using the checkboxes in front of each rule, if necessary. Note that coloring is a view-related property, so you can have e.g. the left pane highlighted and the right normal. You must turn color coding on/off for each dual pane (or tab) separately.

By default, color coding is only shown in the **active** folder pane, to minimize distractions and help you focus. If you nevertheless want to see colors in the inactive pane as well, tick the [advanced option](#) "Filename color coding applied to inactive views..."

Color coding is a resource intensive feature, as every folder you browse must be scanned for matching items to be colored (color [tagging](#) on the other hand is quick and easy as it applies to individual files). Thus you should avoid color rules that use **slow** properties — don't even think about searching for text in files, it will grind xplorer<sup>2</sup> to a halt! Fast rules based on wildcards or simple properties like sizes and dates don't pose a problem. Use the arrow toolbar buttons [2] to push any slow rules to the end of the list.

## Multi-folder containers

Scrap containers were once a novel feature introduced by xplorer<sup>2</sup> ages ago, before other similar tools copied the idea. Regardless they are a very useful concept that allow you to manage files from different folders simultaneously.

Use **Window > Scrap container** menu (from the *main* window) to open a scrap window. Many commands that deal with subfolders (e.g. file search) also create scrap windows automatically to present their results, files that reside in different folders. As you can see, scrap windows look similar to the main window (see [figure 1](#)) and they work more or less the same way. The figure below demonstrates all the possible window features, most of which are optional and can be turned off using **View** menu.

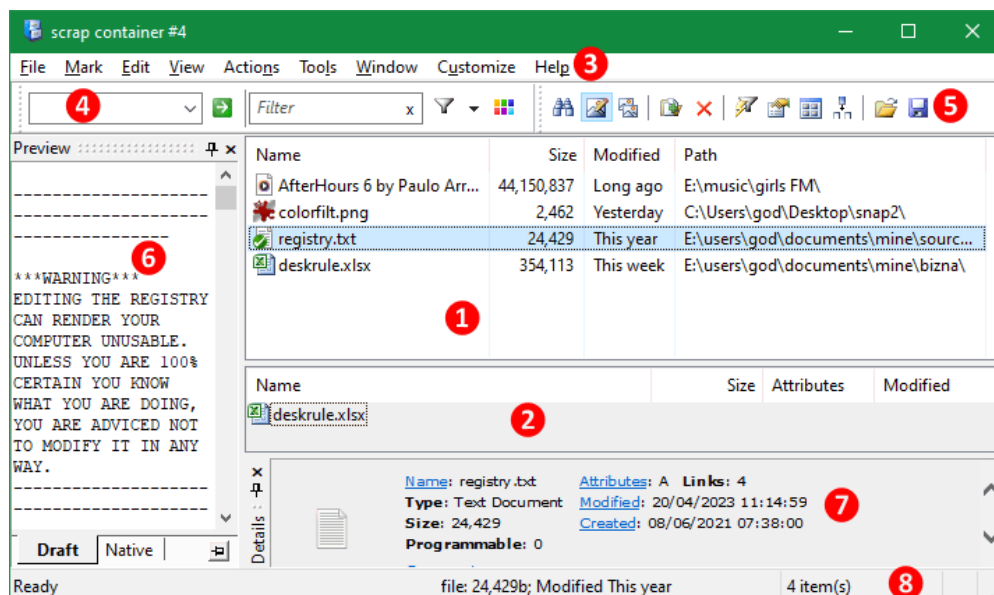


Figure 21. Scrap container window

Here is a brief description of scrap window elements, by their reference number in the red circle:

- ① **Active pane.** In here you hold a collection of items, both files and folders. Notice the **Path** column that shows the full path of each item, demonstrating they are coming from different folders. Click to select one or more items, then apply one of the available menu commands. Notice that there's no titlebar in the active pane as there's no concept of "active folder".
- ② **Second pane.** You don't usually need dual pane functionality in scraps, but it is possible to have a spare pane for [comparisons](#). Turn it on/off using **View > Dual pane** menu and tile it vertically or horizontally. The inactive pane shows with a darker background color. This pane isn't sentenced to permanent inactivity though, just click on it to make it the active pane!
- ③ **Menu bar.** It is organized in a similar fashion as the main window menu so you can find your bearings. You will find almost all the commands there — even a few unique for scrap containers.
- ④ **Address bar.** Here you don't select folders to browse but you could type [commands](#) to execute. The most useful part is the [quick filter](#) box that lets you filter and select items in the active pane — but quick *searching* menu command doesn't make sense and is disabled.
- ⑤ **Main toolbar.** Shows a list of frequently used commands. Use the context menu to set its appearance and customize it adding or removing buttons.
- ⑥ **Preview pane.** This resizable and dockable pane shows a [preview](#) of the `[focused]` item in the active pane, depending on its type. Turn it on/off using **View > Quick viewer** menu.
- ⑦ **Details pane.** Shows extended and [type-related](#) information about the focused item. Turn it on with **View > Details pane** menu, and autohide or close it using the pin and [X] button on its titlebar.
- ⑧ **Status bar.** Shows information about the focused item, total selection byte size, progress and error messages etc. Compared to the [main window](#) statusbar a few panes are missing, e.g. you don't see free drive space.

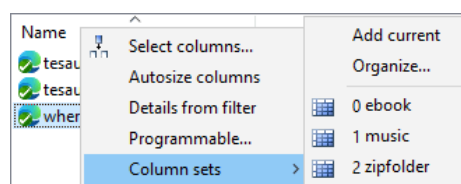
Most conspicuous is the absence of the folder tree and drive bar (see [figure 1](#)), as here we don't *browse* folders individually, but work with item collections. Other than populating the active pane, the workflow is almost identical:

- Set the active [pane style](#) to any suitable mode (details, thumbnails etc)
- Sort, group, filter and select items for further file management
- Use the peripheral panes [6-8] or [peek preview](#) to examine the selected items.
- Use drag-drop and the shell context menu for simple file operations

As long as you have "Save program state on exit" option enabled, scrap containers remember the window size, layout and file details you used last time, and they will look exactly the same next time you open a scrap window. You will probably want to define a few custom [column sets](#) and switch details according to the type of files you work with. You can easily switch column sets if you right click on the view **header** row (see side picture)

**Details from filter** is an interesting menu command you can use if you do a file [search](#) that includes some unusual property like **Author** — or something that's missing from the current details. It checks the hyperfilter last used and makes sure all its properties appear in the active pane, so you can understand the search results.


**Programmable** menu command is an ultimate-only feature that lets you create your own [composite](#) file properties




## Populating scrap panes

You can hand pick items and create your collection in a number of easy ways, bringing items from the **main** xplorer<sup>2</sup> window to a scrap pane:

- Drag-drop one or more items from a folder view or even the desktop
- Use **File > Send to scrap** menu command to insert the selected items (or use <Ctrl+S> keys)
- Copy source items with <CTRL+C>, then "paste" them in a scrap pane — this will merely place them in the collection.
- Copy then paste with <CTRL+V> full paths (in text form); to import many items, there should be just one *full* path per line in the clipboard, no delimiters or quotes.

 When you place items in scrap containers, their actual disk location remains unchanged. The scrap collection is merely taking notes of items' locations. On the other hand items in scraps are real enough and fully operational. For instance if you place a *folder* item in a scrap, it is a live drop target: if you drag-drop *another* item you will have it *moved* (or copied) inside said folder! When inserting items with drag-drop be careful of any [drop targets](#) highlighted and steer clear. Find some empty spot to drop the new items (wait till you see [link](#) drop feedback, which will insert the item without any drama). For safety don't add items with drag-drop altogether, use one of the alternative methods or use *right* mouse button drag.

You can build collections in one go or step by step. Each new batch of items is added to the existing collection. To remove one or more selected items use **File > Remove** menu, or right click someplace empty and use **Clear** from the pane's background menu — that will clear all contents and you can start a new collection.

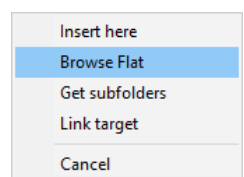
 Removing items from a scrap does **not** delete the actual files, they are just removed from the collection. <DEL> key in scraps merely removes selected items from view; if you want to really delete them, use **File > Delete** menu or press <CTRL+DEL> key combination.

Scrap containers accept items from both filesystem and "virtual" folders. If you want to see file details in mixed-bag collections, say regular files and compressed zipfolder items, prefer [stock](#) columns that work consistently in both types of folders.

## The flat earth view

Moving on from manual to **automatic** ways to populate scraps, there are a few ways to load the contents of a folder and all its *subfolders* in a single scrap window. Such flat views comprise entirely of **files** — there's no point managing the subfolders objects themselves in this situation.

- Use **File > Browse flat** on a folder from the main window; this will open a new scrap container for the contents
- *Right-drag* a folder and drop it in a scrap, then use **Browse flat** menu command; this way you can flatten folders in an *existing* scrap container.
- If you want contained *subfolders* only instead of files, right drag and use **Get subfolders** menu command
- The right drag menu command **Link target** inserts a local *shortcut* to a network item instead of the remote item; see the discussion in [mini-scrap](#) section below
- Highlight a folder in a *dual-pane* scrap and press <Alt+Return> to flatten it in the inactive pane.



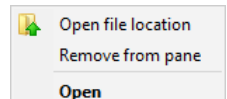
Such flat views are useful when you want to operate on items *recursively* into subfolders, instead of doing things one folder at a time. We are sure you'll discover many more uses yourselves!


## Managing collections

Scrap containers are fully fledged file manager windows, so you can work with them **directly**, without worrying about their container folders. Whether you manually collected the items, recursively flattened folders or browsing the results of a file search, you can preview, open and use the same palette of [simple](#) and [advanced](#) file management commands. Scrap containers have some unique commands too, not available in normal browser windows. You can check for [duplicate](#) files and compare the content of folder [hierarchies](#), find [similar photos](#) and more.

In case you need to open the folder that contains an item in a normal xplorer<sup>2</sup> window, use <Ctrl+Enter> key or right click on the item and pick **Open file location** command from the menu.

If you select items from many folders (especially if multiple file types are selected) the shell context menu works, but it may include too many commands and be very sluggish. Try to use commands from the main menu bar instead, if you can find an equivalent command — most of the basic commands appear both in the main and context menus.




 There are some complications in multi-folder views. If you copy items belonging to different folders, then try to **paste** them somewhere, they will be placed in a *single* target folder, destroying the relative folder structure. Perhaps that's what you need, but in case you want to keep their original path relationship use **Edit > Paste special > Structured scrap clips** menu command. See the discussion in [advanced paste](#) operations.

## Shell "documents"

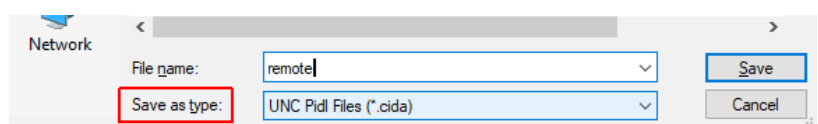
The term "scrap" suggests something fleeting and impermanent, and indeed these containers were originally conceived as quick-and-dirty temporary holders of items, which would be discarded as soon as some simple file operation was performed on them. However, it was soon realized that there was another possible use mode: **saving** scrap contents for future use.

**Actions > Write contents** creates a file with .CIDA extension and writes location information for all the items in a scrap pane. You can reopen such saved files at a later time and restore the scrap container, just like as if it was a shell *document*. xplorer<sup>2</sup> registers .CIDA files for its own use so you can open them by merely double-clicking on them, even when xplorer<sup>2</sup> isn't running — or use **Actions > Load contents** menu.

You can take advantage of this functionality in many ways. For instance you can create cida files as "table of contents" that keep the locations of files with similar content throughout your filesystem. This would be equivalent to creating a normal folder and keeping *shortcuts* to the same documents, but if you go the scrap way you have *direct* view of the actual files. Or you can create playlists out of your music collection and save each as a CIDA file, one for each genre. The sky is the limit!

 We stress that only the *location* of the files is stored in .cida files, not the files themselves. If you delete or move the actual file, xplorer<sup>2</sup> will not be able to reload it when reading the .cida file. When such errors occur you can optionally see a log of all missing files that may help a subsequent manual repair.

Normally you will be reusing cida files on a single computer to access a set of related files. If you want to access the same files from a *different* PC you must save some extra information to make the cida file location-independent; that is the purpose of "UNC pidl files" type in **Save as type** drop-down box, as in this snapshot:




## Multiple scrap containers

At present scrap windows don't support tabbed browsing, but you can open multiple scrap containers for the same functionality (albeit not as screen-tidy). When you have many scraps open, the *last one* you used is considered "closer" than the rest, so if you **Send to scrap** from the main window, the selected items will go to this "nearby" scrap. The target scrap's **titlebar** can be seen on the main statusbar:

[2 item\(s\) added in pipa collection](#)

To be 100% sure of the destination scrap, use drag-drop instead of <CTRL+S>

You can see a list of all windows in xplorer<sup>2</sup> process (including scraps) via **Window > List** menu. To help you identify which window is which, scraps that are generated automatically have a descriptive titlebar, like the one below created by **Browse flat** command:

 New Folder @ C:\subfolder contents

Loading CIDA and search results also have good titlebars; for the scrap windows you open *manually*, use **Window > Set title** menu command to set a titlebar that reflects the contents — that should help you remember what each scrap window is all about.

### Mini scrap pane ▶ PLAY ▶

It can be argued that in some situations it will be more convenient to have a scrap pane (virtual folder, drop box, "playlist", multi-folder view or however you want to call it) *within* the main xplorer<sup>2</sup> window, rather than opening a separate scrap container. This functionality is furnished by **View > Mini scrap** menu command.

There is only one such pane available per window, but it is quite versatile. It is resizable and [dockable](#) on any window side. It is almost identical in functionality to a stand-alone scrap container: you can put in it files and folders from many locations, you can select view modes, sort orders, even arrange the content in [groups](#). Most of the menu commands are available too, including load/save through the background context menu. When you click on the miniscrap, it becomes the *active view* (instead of the folder panes), and you can preview items as you select them in the miniscrap!

The miniscrap is rather small for managing lots of items, but here are a few possible uses:

- Temporarily hold files, tagging them for later processing en masse.
- **Bookmark** pane: when you double-click in a folder (or folder shortcut) in the mini-scrap, it is loaded in the active pane. This can be considered a replacement for the tree, showing only the folders you are interested in instead of the whole hierarchy.
- **Launch pad**: put in some favorite documents or even program shortcuts for easy opening. Program icons behave as drop targets so that you can pass file arguments using drag-drop.

If there is no stand-alone scrap container open, <CTRL+S> sends selected items to the miniscrap. By default the contents of the miniscrap are automatically saved and re-loaded when you restart xplorer<sup>2</sup>; this makes its use as a bookmark pane very attractive and convenient. You can access a number of recently used "playlists" too through the context menu. An alternative is to *right-drag* a stored CIDA file and drop it onto the mini-scrap, picking **Browse flat** from the context menu.

Mini-Scrap

Miscellaneous

☐ Disable auto-refresh of contents

☐ Always reminder to save content, even if not saved b...

☒ Load last contents file (CIDA) automatically on startup


☐ Never ask to save pane content, discard changes


☒ Open folders in the same xplorer2 window

☒ Auto-save scrap pane contents

Miscellaneous

Panel behavior options (CLVOPT\_XXX).

 To stop the mini scrap reloading the last CIDA document on startup, empty its content before you close it, e.g. select all and then hit <Del> — or use **Clear** from its background menu. Or untick the [advanced option](#) "Load last contents file...". For more tweaks see the miniscrap section of the advanced options editor. The default contents file name is %APPDATA%\zabkat\x2\_mvtmp\_.cida


 If you store network items in the miniscrap, then chances are that next time you restart xplorer<sup>2</sup> without a network connection, such items will be considered "gone" and removed from the pane permanently. Instead of adding the real network file or folder, right-drag it and pick **Link target** from the drop menu. This will create a LNK shortcut file in the %TEMP% folder, and add *it* in the pane instead of the remote item. This shortcut will behave the same invoking the network item when double-clicked, but as it is stored locally it doesn't risk being removed automatically. Another use of adding links — even for local items — is that you can **rename** the LNK shortcut and give it a more descriptive name, without affecting the real target. Storing LNK shortcuts in %TEMP% folder isn't robust against disk cleanups; for safety create your own shortcuts somewhere safe instead and emulate **Link target** function manually

One advantage of the [quick access](#) list is that it won't suffer from similar problems, you can add network items in it without issues.



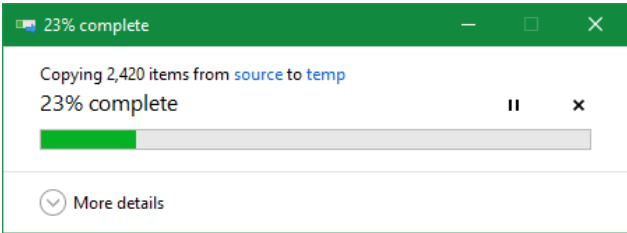
## Advanced file management

Now you have found and browsed the folder you want (or collected items from many folders in a scrap), focused your attention with filtering and selected the files you want to manipulate, it is time to unleash the arsenal of xplorer<sup>2</sup> robust file management tools that can deal with thousands of files in one go.


 Full scale file management is available for regular filesystem folders stored on your hard disks and externally attached and network drives. Special folders like compressed ZIP archives and android devices can be browsed normally, but they are limited to a few [simple](#) file operations. Unavailable menu commands will be disabled.

## Advanced copy/paste

xplorer<sup>2</sup> uses exactly the same clipboard formats as windows explorer so you can transfer files to and from the desktop, external explorer windows and any xplorer<sup>2</sup> windows seamlessly. You begin by copying (or cutting) the source files, browsing the destination folder — the dual pane feature comes handy — then pasting to copy (or move) the files to their new location. Paste operations use the same copy engine as windows explorer (and so does drag-drop), so you will see the familiar copy progress dialog:



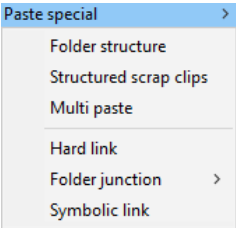
A common operation is creating a duplicate of a file in the same folder. This can be done with two keystrokes, <CTRL+C> followed by <CTRL+V> will paste a "Copy of" item. xplorer<sup>2</sup> offers **Edit > Duplicate** menu command as an alternative.

 When you copy/paste a folder, then it is copied in its entirety, including its files and all **subfolders** it may contain, recursively. **Edit > Paste** menu works even when no files are copied in the clipboard. If you copy some *text* then paste it, a text file is created in the current folder called X2SCRAP.TXT containing the clipboard text. Likewise if you copy an image you can paste it as a file called X2SCRAP.PNG.

Instead of pasting the whole file, you can create a windows shortcut (LNK) file using **Paste link** menu (or while [dropping](#)). These are small files that point to the real file (think of all the shortcut program icons on your desktop). When you double click on a shortcut, the target file is opened.

Other special kinds of targets can be created using **Edit > Paste special** submenu as below:


- If you copy a *folder* item, then **Folder structure** command enables you to paste only the local hierarchy of subfolders to a different location.
- When you copy items from [scrap containers](#) and they happen to reside in different folders, the oddly named **Structured scrap clips** command will paste preserving the *relative* folder structure of the source items - creating subfolders as necessary. In contrast, a plain Paste would place all items in the same folder.
- If you want to paste a copied selection on *multiple* folders use **Multi paste** command, after *selecting* the folders you want to "drop" into. Then each target folder will receive a separate copy of the source files. This feature is more effective in scrap containers where the target folders can be arbitrarily located.
- The remaining 3 paste special menu commands create various advanced NTFS links, and are discussed below.




## NTFS links

The older FAT file system can only do LNK shortcut files, but on NTFS you can create hard links and junction points that effectively impersonate their targets. For example if you open a hard linked file, most programs cannot tell the difference of the link and the "original". Hard links are meant for *files*, junction points for *folders* and symbolic links can do both. To create these advanced links, first **copy** the *target* item, then use the respective **Paste special** submenu (see above).

- **Hard links** create "shadow copies" of files on a single disk partition. You can spread the same file in multiple folders, in a space efficient manner (links take next to no space to store). When you change one instance *all* the other instances are automatically updated. The stock column [Links](#) will tell you how many hard links exist for a file.

 Unfortunately some programs are known to **destroy** hard links when saving file updates. Double check your editor before you start using hard links. If you never edit your linked files there's no such risk.
- **Folder junctions** can reorganize the structure of your filesystem. For example if MS Outlook wants to save its data in C:\appdata\outlook and you run out of space on C: drive, you can put the real folder on E:\outlook and replace the original folder with a junction of the same name (which points to E:\outlook). Then both MS Outlook and your hard disk will be happy! Or you can turn folders that are far away from each other into **siblings** by pasting junctions in a common folder. Junctions can have any name you like although when first created they are called Junction to XXX.
- **Symbolic links** are the equivalent of UNIX soft links and are quite flexible redirecting to both files and folders. They behave somewhat like hard links, if you double click on a symlink you actually edit the *original* file; however if the target is moved or renamed the link will be broken. This feature is for windows vista and later only; you must be running elevated to full **administrator** privileges to create a symbolic link — xplorer<sup>2</sup> will offer to launch itself as administrator to help you with this obstacle. Symbolic links are the only junctions that can point to remote network folders as well as local content.

You can identify junctions by the J file attribute and the little **arrow** icon overlay that all windows shortcuts show. Symbolic links show as having size of 0 bytes. If you select any type of link or shortcut, **Go to > Find target** menu command will jump to the real object pointed at by the link. To see where a link points to use the stock column [link target](#); this information appears automatically in [details pane](#) when a shortcut item is focused.

 When you place any kind of link or junction in a [scrap](#) window, <CTRL+L> will **replace** the link item with its actual target. In case of hard links, it will also find all the remaining instances of the same content! (resolve hard links)


So which kind of link is best? There is no clear cut winner. See the table below for advantages and disadvantages of each type:

Table 2. Comparison of links and junction points

--	--

	Hard link	Junction	Symbolic
File targets	✓	—	✓
Folder targets	—	✓	✓
Network targets	—	—	✓
May jump partitions	—	✓	✓
Elevation required for creation	—	—	✓
Robust against target move/rename	✓	—	—

As you can see for most uses a symbolic link is the best option (if you don't mind the elevation to create one), except for when you rename the linked file. The humble LNK shortcut **can** survive if you rename or move its target, because of the link tracking windows service!

 You can check the formatting of disks from **ThisPC** folder, under **File system** column. Most internal and external hard disks are NTFS formatted and have all filesystem mod cons like junctions and alternate data [streams](#). Only some USB sticks (flash drives) are FAT32 formatted.

#### Further reading

• Redirect MS Outlook PST store folder to a different partition using [junctions](#)

## Robust file transfer

When it comes to transferring large amount of files and folders, the windows copy dialog comes rather short. xplorer<sup>2</sup> in exchange for the fancy animations, offers a robust alternative when copying from and to regular filesystem folders. The main features are:

- Easy selection of destination folder
- Copy all files or those that match a filter
- Parallel or queued transfers
- Overwrite control for backup copies
- Many copy options (e.g. silent mode)

One big advantage of dual pane file management is that the inactive pane is a natural and convenient target for copy/move operations. Instead of copying then pasting, you select the files and use **Edit > Copy to** (or **Move to**) menu commands.

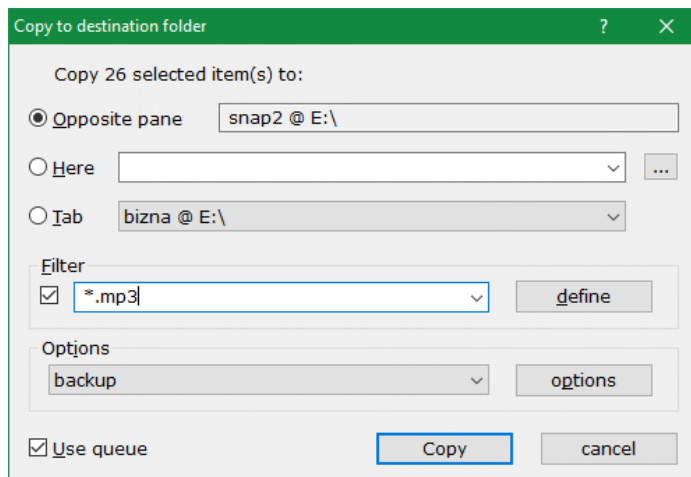


Figure 22. Copy to destination dialog

The dialog has 3 main areas: *where* you want the files copied, whether to apply a *filter* and various *options*.

- **Destination.** If you have dual panes, **Opposite pane** option is selected, which will send the files to the *inactive* pane. If you have any folder [tabs](#) in the active pane, you can pick a **Tab** as destination (choose from the drop-down list). Finally you can type or browse a folder's path in **Here** edit box, if you prefer to pick a completely different folder. These 3 destination types are mutually exclusive, hence they are selected by *radio buttons*.
- **Copy filter.** When you copy whole subfolders, you can choose to transfer *only* files that match a filter. The **Filter** input box understands both simple [wildcards](#) as \*.mp3 (only MP3 files copied) or precise [hyper-filters](#). The drop-down portion lists filters you've saved earlier, or click **Define** button to design a property based filter on the fly. Make sure the filter checkbox is **ticked** to use the filtering function (this needs to be ticked every time you reopen the dialog). If the checkbox is clear, *all* files will be copied



Filtering is applied to **files**, whether to copy them or not. All folders and subfolders are entered recursively regardless of filtering.

- **Transfer options.** There are many [options](#) for overwrite control, various confirmations and error reporting, and task **queuing**. These will be explained shortly. You can quickly select default or frequently used (saved) options from the drop-down list, or click **Options** button to select custom ones.


Once you have all your copy parameters set, click **Copy** button to start the work — or **cancel** to abort. If you used **Move to** command, the action button will read **Move**.



When you are *moving* items to a folder in the same **partition** (e.g. from C:\TEMP to C:\NEW) the command is served by windows explorer, since moving within partitions it very fast (if there is time to show a progress window, you will recognize it as that in drag-drop operations). If you use a move *filter* though, then xplorer<sup>2</sup> will handle it.

Most of the time you don't need to tweak any copy options. You just press <F5> key to show the dialog and immediately press <ENTER> to start the robust copy using whatever options you used last time. In fact to copy to the inactive pane with the last used options press <CTRL+F5> key combination, and it will be done without showing the dialog!

#### Further reading

• Copy specific files with filters [demo video](#) 

## Copy progress

File transfers usually take some time to complete. A progress window is shown, but you don't really have to sit and look at it. Your attention is only required in case of errors and confirmations, which will interrupt the task and show a message box.

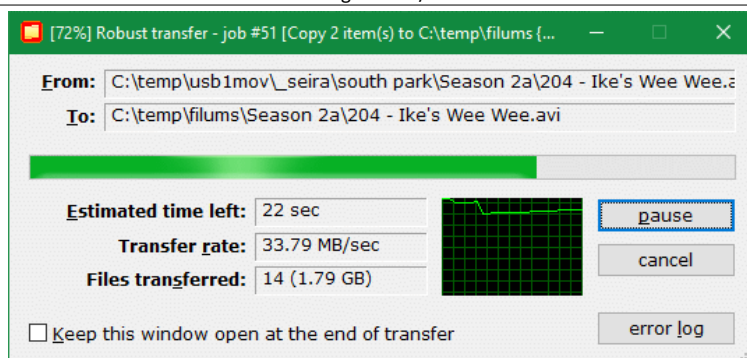
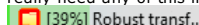



Figure 23. Copy progress dialog

The progress dialog is horizontally-resizable. You can see information about the file being copied (**From** and **To** fields), a progressbar indicating how much work is done, a rough **Estimated time left**, and some statistics associated with the transfer; there's even a **graph** of the transfer speed over time, to keep you mildly amused :) — you don't really need any of this information. You can casually inspect the progress also on the (windows) taskbar icon:



 The percent progress and estimated time left are **not** very accurate, especially when copying subfolders or skipping lots of files in backup mode. Knowing how long a task is going to take won't make it complete any faster! If you want accurate estimates, tick the [option](#) "Calculate total file size upfront" — but this may delay the start of the transfer if there are many files/folders selected.

A **pause** button exists to temporarily halt the file copy (for whatever reason); then click on **resume** button to continue, or **cancel** to abort the copy altogether. If you abort, the files transferred up to that point are not deleted. Whatever's done, is done. If you want to **restart** a failed or abandoned copy, use "Overwrite if newer else skip" option (see below) so that you don't waste time with files that were successfully copied in the first attempt.

If the source and destination folders contain items with same names, they will be **overwritten**. The old version will be replaced with the new file being copied. As this is a radical and irreversible step, by default xplorer<sup>2</sup> requests your approval with this messagebox:

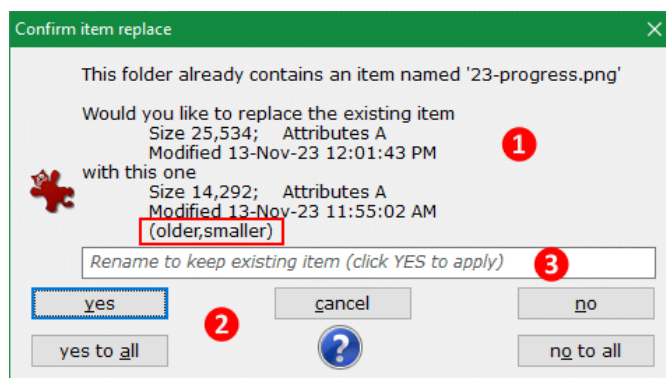


Figure 24. Confirm item replace dialog

To help you decide which file to keep, xplorer<sup>2</sup> shows brief information (size and date attributes [1]) about the *existing* item in the destination folder, and the *new* one that is about to overwrite the old version. You should focus on the last line that highlights the **differences** of the new item **older,smaller** and then decide: to overwrite the old file click **Yes** button, or **No** to keep the old (existing) file. The action buttons [2] include yes/no **to all** buttons, that reply yes or no to any subsequent overwrite confirmation automatically. You see the pattern in the first few confirmations and you decide whether the remaining can be dismissed in the same fashion. If you click **cancel** at this point you abort the entire file copy.

Another option is to keep **both** files, saving the new one with a different name you specify in input box [3]. Doing this file by file manually can get tiresome, but you can tick the [option](#) "Rename target" which will create such unique name copies automatically (it adds **\_1** to the base name). However keeping all these extra files isn't sustainable in the long term, sooner or later you must decide if they are all worth keeping or are just **duplicates**.

When it comes to replacing **folders**, i.e. when the destination contains a folder with the same name as the source, by default the target isn't *overwritten* — just extra files are added in it. If you want to use the same overwrite semantics for both files and folders, tick "Replace the entire folder" option; then if a folder collision is detected, the entire old folder (cum subfolders) is first deleted, then replaced with the new contents. This is liable to destroy a *lot* of old content, so please be careful.

For most people, the smarter overwrite option is "Overwrite if newer else skip" (available as **backup** from the drop-down list, see [figure 22](#)). It is an *automatic* way to resolve conflicts; it compares file **modification** dates and keeps the most recent file (assuming it contains later changes you made). If the source file is older or identical (in terms of date modified) to the existing, it is **skipped**. This mode is perfect for **backups**, where you keep an up-to-date copy of your working documents in a safe external location (e.g. USB disk). If you do this daily, you don't need to copy your entire documents folder, only new and changed items. Thus you save time and effort — no time wasted confirming file replace operations.

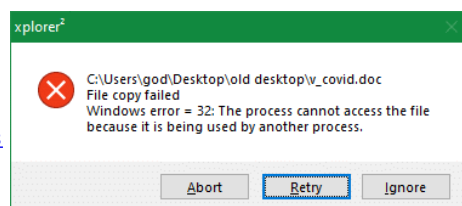


A convenient shortcut for backup operations is **Tools > Backup copy** menu command. You browse your working folder in the active pane and the backup location in the *inactive*. The command will select all files and folders in the active pane and copy them to the inactive pane (e.g. an external drive), using "Overwrite if newer else skip" copy method — and in the end it leaves your default copy options unchanged

## Error handling

There are a number of situations where a file copy fails, as for example if the target file is open in an application. xplorer<sup>2</sup> will pop a message describing the issue, and depending on the situation will offer alternative courses of action, e.g. **Abort-Retry-Ignore** questions, where you click the button with your choice action. Usually the only feasible options you have are to **ignore** the problem if it looks like a 1-off, or **abort** the copy operation if more files are expected to fail in the same manner — say if you ran out of space in the destination drive.

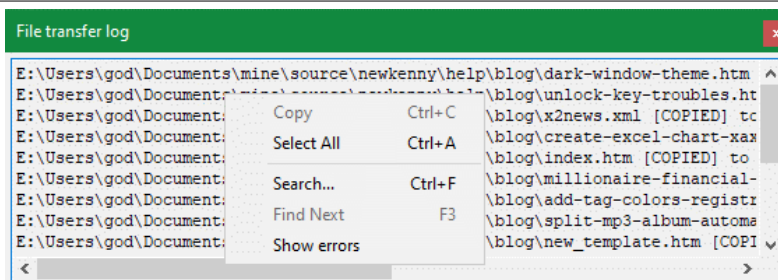
A soft error you may encounter copying to FAT32 devices is about "Secondary streams" that couldn't be copied. [ADS](#) are used for storing comments or to mark downloaded files as *risque*. If you are certain you don't need this warning message (if you are not using file comments) tick the [advanced option](#) "Disable stream loss warning..." — one less thing to worry about.




The progress dialog has a **log** that records each file operation individually. You can browse this using **error log** button (see [figure 23](#)) — a misnomer really because it shows more than errors. You can search the log pressing <CTRL+F> keys, or if you just need the error lines, right click in the log window and pick **Show errors** menu command. For subtle problems search for the special @@@ marker.


When you copy in backup mode, only files that were copied are logged (newer and modified) — there are usually fewer of them in large backups.

To make sure you get a chance to see the log after the end of the transfers, tick the [option](#) "Keep progress window open..."



 Under normal circumstances, robust copy does not work *into* protected operating system folders like Program Files. Either you need to start xplorer<sup>2</sup> elevated (**Window > Administrator** menu), or even simpler, use **drag-drop** to copy the files — this is automatically elevated after a UAC prompt authorization.

#### Further reading

- ▢ Bypass UAC restrictions to copy into [system folders](#) 

## Unattended file transfers

When you need to copy terabytes of data you may leave xplorer<sup>2</sup> running overnight to finish the job. Imagine waking up in the morning to find xplorer<sup>2</sup> stuck half-way into the unfinished job, showing you an error message! To keep your blood pressure down use "Silent" option for this scenario. xplorer<sup>2</sup> will work showing just the progress bar, and won't bother telling you about errors or asking you for overwrite confirmations. When is all said and done, if any problems were encountered, you get a chance to see the error log — but *after* the copy is finished one way or another.

During silent operation, any events encountered that would normally require user intervention are dismissed automatically. More specifically:

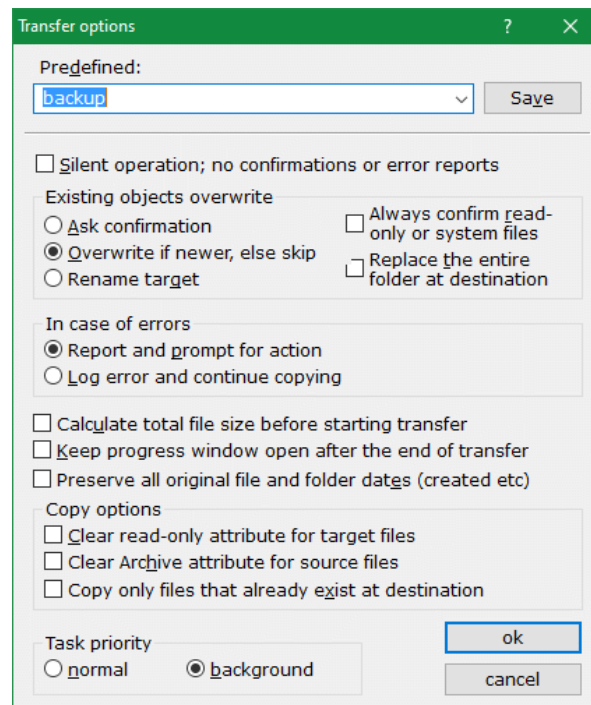
- all errors are ignored, and any problematic files are skipped (you can read about these in the log)
- all special confirmations (e.g. for replacing system files) are allowed — "yes to all" answer implied
- all files are replaced automatically, just as if you answered "yes to all" for overwriting

If you are doing unattended **backups**, then tick "Overwrite if newer else skip" before you tick the silent option. Then you get the benefit of skipping files that are not newer, copying only the files necessary for the backup.

## Transfer options

The robust transfer configuration [dialog](#) has a button for **Options**. Many important options have been discussed already; here's a brief explanation of all copy options:

- **Silent operation.** Prevents any messages or interaction until the copy is finished, then shows any errors in the end. If ticked, the following 2 checkbox groups are disabled.
- **Existing objects overwrite.** Pick one of the 3 available alternatives: ask for confirmations for each file, automatically overwrite new and modified files (backup mode), or keep both files (the new one is renamed)
- **Always confirm readonly & system.** Special attribute files are always confirmed even in backup mode
- **Replace the entire folder.** Folders behave like files on collision; the old contents are erased completely before overwriting
- **Error reporting.** Either you see errors immediately as they occur and decide how to proceed, or they are ignored silently skipping any problematic files
- **Calculate total file size.** If you want accurate remaining time estimates tick this option — it may delay the start of the operation
- **Keep progress window open.** Gives you a chance to examine the error log at the end of the transfer
- **Preserve all dates.** By default, when a file is copied, its modification date remains the same but the *creation* date may be newer (!) Tick this option to ensure that the copied files have the same modified **and** created timestamps, even when new files are created at destination
- **Copy options.** Some legacy attribute manipulation (for copy, *not* move operations) that are probably irrelevant nowadays, like clearing the read-only attribute (if copying from CDROM) and clearing archive status (backups no longer rely on A attribute)
- **Copy only existing files.** With this option files that exist in the source folder but not in the destination folder, are skipped.
- **Task priority.** Better set this to *background* so you can use your computer during the file transfer. Set to *normal* only if you want maximum speed, but your PC may become sluggish.



Set of options can be **saved** and reused. First tick all the options you need, then give them a descriptive name in **Predefined** box, and click **Save** button. Then you can quickly select saved option sets straight from the main transfer dialog (from the [options](#) drop-down list). To modify a previously saved option set, change the tickboxes and click **Save** again (using the same descriptive name) to overwrite the old definition.

## To queue or not to queue?

xplorer<sup>2</sup> can happily furnish multiple copy operations in parallel, each in its own thread, however this may *slow* things down, especially if you are copying from or to the same hard disk units. Instead of thrashing your hard disk with parallel operations you can have them **queued** so that they are completed one after the other. Tick **Use queue** option in [copy to](#) dialog and leave it ticked for all future copies.

If xplorer<sup>2</sup> is in the middle of a large copy, any extra jobs will be added to the queue. You don't have to wait, just use **Edit > Copy to** command normally, and once one job is finished the next one will begin, until the queue is empty.

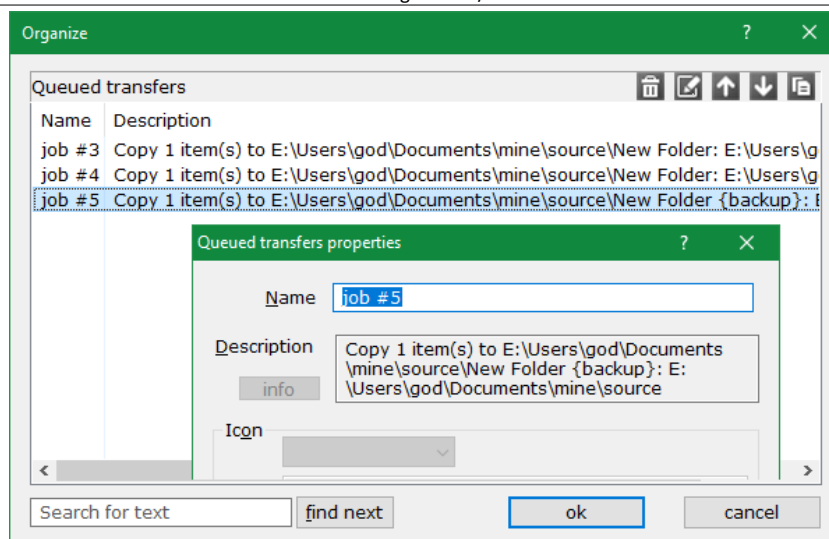


Figure 25. Copy queue management

You don't normally worry about queued copies, but if you want to see what is waiting use **Edit > Queue status** menu command, which will show you queued jobs in a dialog similar to bookmark [organization](#). You cannot change properties of any waiting jobs, but you can delete or move them up/down the order using the dialog toolbar buttons. Double click to see details on the source/target folders and any special options used.

Another recommended [option](#) is to set the task priority to **background**, which won't overwhelm your PC resources, thus you can continue working during big copy tasks.

If you opt for **silent** operation, then no errors will be shown till *all* queued jobs are finished. In the very end, and only if there were any problems, you will be notified about it.

### Partial support for special folders

The full power of robust copy commands is available for all filesystem folders (both local and remote), where you work more often. In other special folders that hold "files" of sorts, as compressed zipfolders or mobile phones, there is partial support for some copy options including "Overwrite if newer else skip" that can be used to **backup** your phone, but only in one direction (*from* your phone to your PC but not vice versa). Filtering isn't supported either, *all* files are copied.

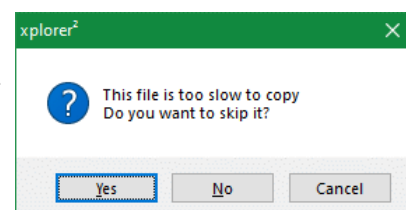
When xplorer<sup>2</sup> is copying from or to [special folders](#) you will see the regular windows explorer progress and overwrite confirmations windows, that look different — compare with the snapshots in this section.

### Advanced copy tweaks

Using the external [advanced options](#) tool, you can tweak the following important copy options:

- **New item highlighting.** When you are copying and the target folder is visible (e.g. the inactive pane), and there's no prior selection in the target folder, then each new file added is shown highlighted (selected and focused). This lets you see which items were just copied — they should be all selected in the end. If *large* files are copied *slowly*, the sensor may not pick up all of them; increasing the "New item sensor timeout" will help in this situations.
- **Large overnight copies.** Setting *Silent* option isn't enough for unattended file transfers. You have to make sure your PC won't go to **sleep** or hibernate before the copy is over. xplorer<sup>2</sup> is doing this by default — will go to sleep eventually when the copy is over, but if you nevertheless *do* want to interrupt the progress tick "Allow system to sleep..." option.

**Copy watchdog.** When you copy to or from slow network locations, xplorer<sup>2</sup> is monitoring the transfer speed. If nothing happens for 20 seconds or so it will alert you to the problem with the message *This file is too slow to copy...* You can either skip the slow file or persevere with the slow copy (in *silent* mode the file is skipped automatically). If you get too many of these interruptions, and your network is known to be slow to respond, you can increase the "Network folder ping" timeout to match your network (e.g. set it to 30000 milliseconds if half minute gaps are normal). Alternatively you can disable the watchdog altogether ticking "Don't check for slow copy..." advanced option.



### Robust deletion ▶ PLAY

Similar to robust copy, xplorer<sup>2</sup> can do robust file and folder deletion. When you delete a lot of files with windows explorer and one happens to be locked or any kind of error is encountered, the operation is aborted mid-way. The robust deletion engine in xplorer<sup>2</sup> is designed to keep on going where windows explorer falters.

Whenever you start a delete operation that cannot go in the [recycle bin](#) (e.g. you press <Shift+Del> or delete in an external flash drive) the robust delete dialog kicks in:

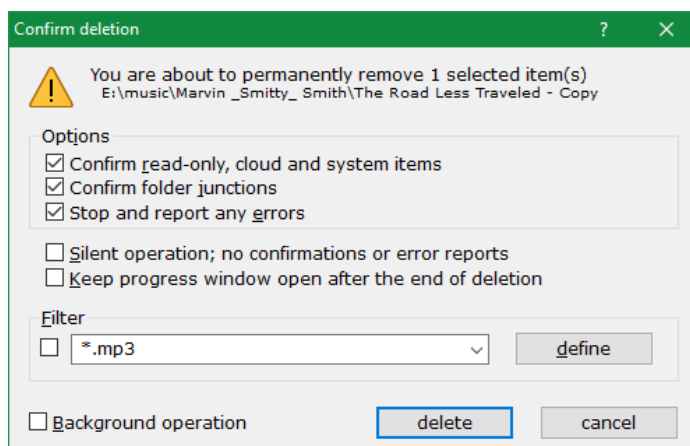





Figure 26. Robust delete options

The first row of the dialog shows **how many** items are going to be deleted (beware of selected items hidden out of view you didn't mean to delete), and the full paths of the first few of them.

Deleting is much simpler than copying, but the robust delete dialog offers some similar options, as confirmation controls (checkboxes) that are self-explanatory. There is also a "Silent operation" option when you do unattended deletions and don't want to be delayed with confirmations or error reports. Errors are ignored but then probably the deletion will be partial.


When deleting **folders**, all their contents are removed recursively down to the deepest subfolder. This includes any hidden or system files that may reside within. If you don't have permission to delete them, an error will result.

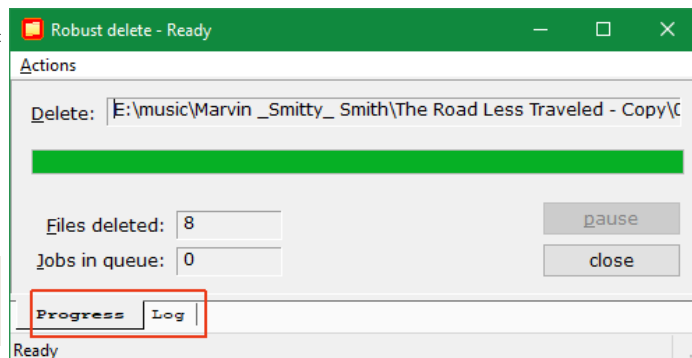
 "Confirm folder junctions" is a legacy option for older windows. Once upon a time deleting junction objects could end up deleting contents of the target folder — but the problem was fixed in later windows. To be 100% safe, when you want to remove the [junction](#) object use **Edit > Paste special > Folder junction > Delete** menu command.

You can even specify a **filter** (either a simple wildcard or a complex [hyperfilter](#)) so that only files that match it will be deleted. This makes sense when you are deleting into subfolders and you only want to remove a certain kind of file.

Click on **Delete** button to start the deletion. You get to see a progress report dialog like the one to the right. It is possible to **pause** and resume deletions, but they are usually too quick and you won't have time to do it! The dialog has a tab control you can switch among displaying progress or the **Log**. The latter text information is searchable, right click to see the search menu (like the one in [robust copy](#), log)

The little **Actions** menu has commands to change the log's font and clear it. There's also a delete queue but usually it is empty unless you do a lot of successive deletions.

 Robust deletion is permanent, deleted files won't go in the recycle bin hence you cannot undelete them easily. Use this command with care — that's why there's no option to skip the robust delete confirmation, it's your safety blanket.



## Mass renaming

Renaming filenames one by one can only work for casual name changes. **File > Mass rename** menu command lets you apply various transformation rules to all selected filenames. Here are some examples of what can be done:

- Add an auto-increasing counter to a list of filenames `name1, name2, ...`
- Insert file properties in filenames, e.g. rename MP3 files according to their artist and song title information
- Search and replace parts of the filename
- Delete parts of the filename
- Change case e.g. CAPITALS or small letters

All this and more can be achieved through this very compact mass rename user interface:

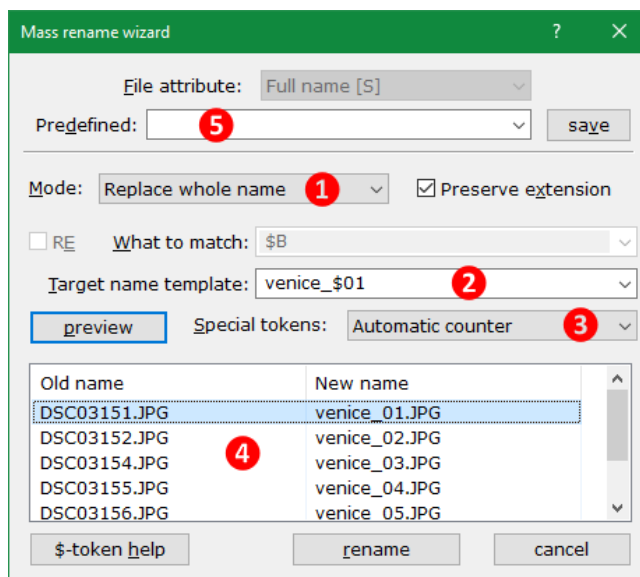
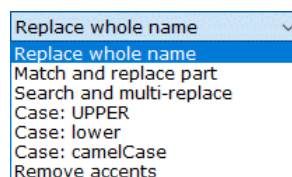



Figure 27. Mass rename dialog


The most important setting is the **Mode**. Using this drop-down list [1] you set various modes that determine what kind of renaming is performed:

- **Replace whole name.** You compose a new file name using **Target name template** box [2]. You add variability to the name template using various special tokens from [3]
- **Match and replace part.** This is a simple search and replace operation in filenames, use **What to match** to set the string to be searched for, and what to replace it with in [2]. Usually you search for something fixed but the replacement template may have variability with \$-tokens if necessary.
- **Search and multi-replace.** Simple match and replace will only do the *first* match of the string you search for (if any exists). In multi-replace mode *all* occurrences are replaced. For example if you search and replace `OLD` with `NEW`, the filename `oldAndold.txt` will end up `newAndold.txt` in single match mode and `newAndnew.txt` in multi-replace mode. Note that search is not case sensitive, `OLD` will also match "old"
- **Change case.** There are various methods to choose. Here you don't supply any template (both search and replace fields [2] are disabled). Each existing filename is CAPITALISED or turned to lowercase; camel case is a curiosity that turns spaces and word breaks into capitals (e.g. "hello world-son.txt" becomes `helloWorldSon.txt`). Finally **remove accents** leaves only plain letters in filenames so the German `über.jpg` will become `uber.jpg`, slovak `žena.png` will turn to simply `zena.png` and the like for all diacritics.




 As discussed already, changing the filename means changing the *title* (aka base name) only, not touching the 3-letter extension that determines the type (eg. .TXT).



 Ticking **Preserve extension** box ensures that any modifications will not touch the all-important extension part — only the base name. In the rare occasions you need to change extensions, untick this box, and beware the usual gotchas.

The list [4] shows you a **preview** of the transformation that is going to happen if you go ahead with the rename. The column **old name** lists a sample of the currently selected filenames in the active pane, and **new name** shows how each name is going to change. xplorer<sup>2</sup> automatically generates a preview from time to time, or you can click on **Preview** button to see the pending name changes. If your names are long you can *resize* the dialog window horizontally or/and vertically.

 Mass renaming allows lots of flexibility with its templates, but this increases the risk of mistakes. Always preview the name changes before you commit the rename, because you may foul the names of many files — *without* automatic undo capability. Preview will catch some common issues as 2 filenames ending up with the same name, and will highlight the problems for you. Any [illegal](#) characters are automatically replaced with underscores.

## Variable rename templates

Mass renaming is helped by the use of special tokens that represent part of the filename and other useful constructs. They all start with a **dollar** \$ sign followed usually by a single letter (e.g. \$B stands for **B**ase filename). These \$-tokens get replaced with what they represent for each filename processed — see for example how the counter token \$01 in [figure 27](#) generated a different number for each filename. Thus you can create compact reusable templates and use them for many different rename jobs.

If you remember the name of the variable token, you can type it straight in **Target name** box [2] or use **Special tokens** [3] list to help you find the token you are after. You can use multiple \$-tokens in a template. The tokens useful for renaming are a subset of the available [\\$-tokens](#) used for command scripts. Here are some examples:

- **Automatic counter.** \$1 will create an incrementing series of numbers starting with 1,2,... You can start with a different number and add leading zeros e.g. \$003 will generate the series 003, 004,... Note the files are assigned numbers in the order they are *shown* in the active file list (the folder you are working in) — make sure you use the appropriate sort mode or even [manual](#) positioning, before starting to rename.
- **Append to name.** You can prepend and append to the existing filename using \$B (base name) token in combination with the text you want to add, e.g. FILE\_\$B will add the prefix FILE\_ to each filename.
- **Change extension.** A template like \$B.DAT will change the extension (type) of the selected files to DAT while keeping the base name. **Preserve extension** tickbox should be cleared to change the entire name including extension — although it is generally unadvisable unless you know what you are doing!
- **File attribute.** Lets you insert any file property using the [column name](#) in curly braces, e.g. the template \${#} \${Album} - \${Title} will create MP3 filenames as 2 Mustang - Fly little bird using their ID3 tag information (track number, album and song title). Make sure all selected files have the properties you use in such templates, otherwise the \${token} will be replaced with *nothing* and you may get the wrong filenames.
- **Character range.** This token can represent part of the filename by the starting letter offset and a range number. The format is \${first:length}, so \${3:2} acting on the name trial.txt will extract 2 characters starting from position 3 (that is "ia"). You can use this in a variety of ways in search and replace mode, e.g. leave the replace (target name) field empty to *remove* that part of the name. Passing 0 for length is equivalent to an *insertion* operator, e.g. searching for \${3:0} and replacing with "add" will insert the string "add" before position 3.
- **Dollar.** What if you want to insert a plain dollar \$ symbol in a filename? Use two dollars \$\$ in a row to escape the special nature of any \$-token.



## Rename with regular expressions

Tick the **RE** box (to the left of **What to match** field) to enable changing filenames with complex [regular expressions](#). This is an advanced renaming mode that can help with certain kinds of rearranging parts of the filename using backreferences. This box is only available when **Mode** [1] is set to search and replace, and disabled in all other modes.

Here is an example. Let's say we have a filename LOG.17112006.TXT that contains a date in DD-MM-YYYY format, and we want to convert it into a more standard YMD (year first) format. We use a regular expression to match the 8 digits separated in 3 groups (day, month and year), then use backreferences to rearrange the 3 groups.

To match the original date format, we use (\d\d)(\d\d)(\d\d\d\d) in the **What to match** dialog box. There are 3 groups defined in brackets that match the date in the filename:

```
(\d\d)(\d\d)(\d\d\d\d)
LOG.17112006.TXT
```

To invert the location of the 3 groups and add separating dashes between the year, month and day, we use \$3-\$2-\$1 in the **Target name template** (replace) box. \$3 is a special backreference that stands for the 3rd group (year), so the resulting filename will be LOG.2006-11-17.TXT

Note that parts of the original filename that were not matched by the regular expression remain unchanged. You can use any kind of search and replace trickery for other tasks too, e.g. to eliminate parts of the filename (leave the replace string empty). Sadly you cannot combine \$-tokens within regular expressions.

## Bulk rename workflow

To rename lots of files in one go, browse the folder they reside and select them (or create a list of items to be renamed in a multi-folder [collection](#)). If you plan to use automatic counter tokens (\$1) make sure you arrange the selected items in order beforehand.

Once you setup the rename parameters and **previewed** the results, click **rename** button to start renaming. Selected files are renamed in the order they are listed in the active pane. A progress bar on the statusbar shows you how much work is left to be done. If everything goes according to plan, you will see the new names at the end of the operation. If there are any problems, you will be informed and asked what to do; if the error is important just abort the rest of the files.

The mass rename dialog has many options. You can save your favorite rename operations, including all options, search and replace strings and so on with a name, and repeat them again later. Use the **Predefined** drop-down box (item [5] in [figure 27](#)) to supply a name for the current options, then click **Save** button to store them. In subsequent runs, use the drop-down portion to access your predefined settings and repeat rename operations easily.

Note the **File attribute** drop-down box is permanently disabled in mass rename mode. This field is used for changing other text attributes (comments etc) [in bulk](#)

## Rename in stages

Sometimes you may need to apply various rename operations to get the filenames in the shape you want, and you cannot do it all in one search and replace step. In this case you must run the mass rename command 2-3 times in succession, e.g. first replace spaces with underscores, then capitalize the names. Mass rename command makes such operations possible by keeping the originally selected files selected, despite the name changes. Then each subsequent <F2> command will work on the same set of files!

Depending on your sort order, the selected files may be rearranged after each rename step, e.g. if you sort by name. If you want to preserve the original order as well (e.g. if you plan to add numbers to them), tick off "automatically resort contents..." [option](#) from Window property page.

This procedure could be automated using [macros](#), especially if each rename step is saved and available from **Predefined** drop-down list

xplorer<sup>2</sup> makes extra effort to keep the original set of files selected and in order, which unfortunately leads to *slower* renames. If you are not using multiple rename steps you can set a small number to "Stepwise rename limit" [advanced tweak](#)


### Further reading

- How to [remove](#) fixed or variable parts of filenames
- Add author and title information in [ebook filenames](#)
- Remove [funny symbols](#) from filenames

## Changing timestamps and attributes

Each file and folder has some basic filesystem properties, its name, its byte size, various dates and attribute flags like Archive and Hidden. These properties change *naturally* as you use, modify and move files around your hard disk structure, but with xplorer<sup>2</sup> you can change them to taste as well.

For files, date **modified** tells the *last* time you edited the file, made some changes in its content; whereas **creation** date is record of when the file was originally created. File dates and times are accurate to the millisecond although you can choose to see [less detail](#).

 It is possible to have the date created *later* than the date modified (!) if you create a copy of the file after you edit it. For folder items, date modified reflects the moment you last added or deleted files in the folder — but only immediately contained in said folder, changes in subfolders aren't tracked at the parent folder level.

File attributes are status flags and are mostly interesting for anorak types :) — see the expert section below. The shell [property sheet](#) allows changing *some* filesystem attributes like **Archive** or **Read-only**, but not all of them. xplorer<sup>2</sup> offers access to all attributes as well as to timestamps of files. Using **Actions > Change attributes** menu command on a selection of items you get this dialog which allows you to change 4 basic attributes and **touch** dates/times.

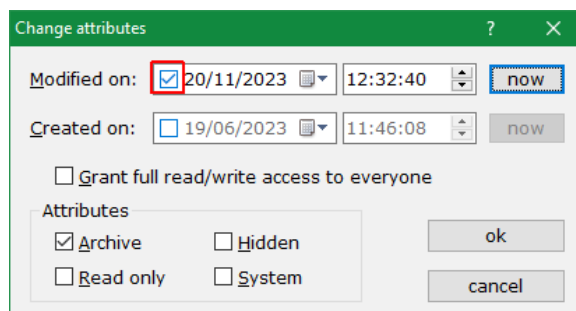




Figure 28. Changing file dates and attributes

When the dialog first appears, it shows the current dates and attribute flags of the selected item. If more than one items are selected, then it shows dates of the **first** one and common attributes are ticked. If only some files have an attribute (and some don't) then the relevant box shows in  **Hidden** indetermined state instead of ticked. Only things that you actually change are affected; so if you don't check any date boxes, these won't be updated. In this way you can change only attributes or just dates, or both at the same time.

To change a date, first tick its checkbox to enable the date and time change controls (observe how the modified row is enabled but date created is *disabled* in the above dialog snapshot). You can then set a specific date and time, or click on **Now** button to reset the date to current time (this is called *touching* a date).

 If you want to bypass file security and grant full access to everybody tick the relevant box. Note that this will erase all prior security information for the selected files and folders so please use it cautiously. If you need to preserve group access permissions then use the windows file property page instead (click on **security** tab and add the Everyone group and assign full access permissions).

When you are all set click **ok** button to go ahead with the changes, or **cancel** to abort the operation. If multiple items are selected, they will all be changed in the same fashion — they will be identical in terms of whatever changes you made. Otherwise each item keeps its original dates or/and attributes. The active folder pane will autorefresh to show you the new attributes you just set for the selection.



## File attribute flags

You may have heard of file attributes like hidden and read-only, but do you know what the **offline** attribute stands for? All will be explained in this section. xplorer<sup>2</sup> attribute column shows up to 12 file attributes; if an attribute is present, it's initial letter is printed, otherwise a dash, e.g:

-HSA-----

means that the file has the hidden, system and archive attributes set, and all others are off. Here's a list of all supported file attributes:

- **Read-only**. Once upon a time you could protect a file against changes and deletions setting it to read-only, but nowadays only a few programs respect it
- **Hidden**. File or folder isn't meant to be seen by people at large, being special or not useful for everyday work. Setting this attribute can [hide](#) items from view.
- **System**. Marks system files and folders like `THUMBS.DB`
- **Archive**. Legacy status flag advertising that file has changed since last backup and is ready for archiving
- **Directory**. Folder items have it, lets you identify folders using the attribute property
- **Compressed**. File is stored compressed taking less space but becoming slower to access
- **Encrypted**. Contents are encrypted so they are not visible by other users
- **Offline**. Marks slow files not immediately accessible e.g. cloud-only remote files
- **Junction point**. Set for [reparse points](#) like folder junctions and symlinks.
- **Not Indexed**. Set for files that should be excluded from search indexing (skip desktop search)
- **Sparse file**. Too [advanced](#) to mention
- **Temporary**. Marks files that are short-lived, usually in `%TEMP%` folder.

Most of these attributes shouldn't be changed directly, or are ineffective (legacy) even if you do change them, so the general principle is not to bother with them. Some special attributes supported for files on NTFS partitions like **Compressed** or **Encrypted** are modifiable through the shell property sheet (**Advanced** button in General page). A few attributes are inherent to the nature of the object and cannot be changed at all, like the **Offline** file and folder **Junction** attributes.

## File date manipulation

Except for *fixing* dates (set the same date for all selected items), you can swap various date properties, and shift a file modification date forward or backward. Use **Actions > Change dates** menu command (or the equivalent ribbon UI command found on **Workbench** tab, under Attributes button) to modify the selected items' dates.

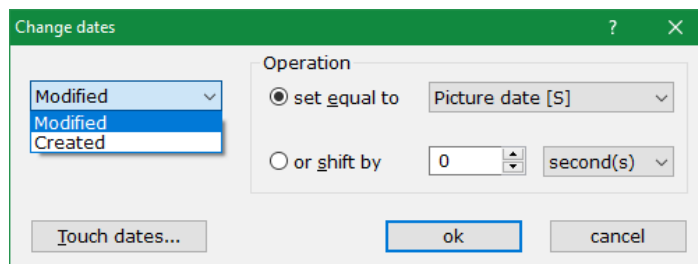




Figure 29. Change file dates dialog

First choose which date to change (Modified or Created) using the leftmost dialog drop-down box, then choose the operation to apply. You either set the date **equal to** some other date property of the same file (e.g. set the modification date to the date a picture was taken or the video media creation date), or you can shift the existing date relatively by some amount forward or backward. Use negative numbers in **Shift by** box to set the date in the past.

It is also possible to set a date from the filename too, if the name contains the standard date format YYYYMMDD HHMMSS (e.g. some cameras set names like PIC\_20181026\_192237.JPG). For this operation select **Name [S]** from the property drop-down box. Naturally trying to set non-existent or invalid dates will generate an error message (e.g. text files don't have a *Picture date* property).


 At present you can only change the modified and creation dates. So you can set the modification date equal to picture date (in case the former got corrupted) but not vice versa. Each selected file ends up with a *different* date, relative to its existing date properties.

A possible application of **Shift by** operation is to adjust modification dates that were adversely affected by summer/winter time changes (DST) that may seem like fake file modifications to backup and [comparison](#) programs. Sliding dates up or down by one hour exactly will cancel the problem (prevalent mostly on FAT32 filesystems). You can shift dates by seconds or years, depending on the unit you select from the drop-down box.

Another way to remedy such phantom DST "modifications" is **Actions > Sync-touch** menu command, that will match filenames in the active folder pane with namesakes in the *inactive* pane, and will transfer date modified information **from** active to inactive panes. For this to work you should have the "original" and "backup" folders in the active and inactive pane respectively, and make sure that all differences are down to DST — or you may forgo *real* modifications in content! Here is a sync-touch [demo video](#) 

Click on **Touch dates** button if you want to set all selected file dates to the *same* date and time — this actually uses the [previous](#) dialog for changing attributes.

## Splitting and merging

 This command isn't as necessary as it was 20 years ago when xplorer<sup>2</sup> first introduced it. Nowadays we don't use space restricted devices like DVDs — or floppy disks! Perhaps it might be useful for sending large email attachments in segments.

Sometimes you may want to transfer large files between two PCs and find that a single file is way too big to fit on e.g. a floppy disk or even CD. The **Split** and **Merge** commands from **Actions** menu will come handy in such a situation.

**Actions > Split** menu expects a *single* big file selected (the command is disabled when you select more than one file). First you break down the large file in as many portions as necessary, declaring your preferred **split size** (in *kB*) in the dialog, The **standards** drop-down lists a few typical media you may choose to provide an automatic split size — e.g. portions that will fit in a single CDRom. **Split pieces** will give you an idea how many chunks you will get for the selected file, for each selected split size.

Then you choose the **target folder** that will receive the fragments and click **ok** button to start splitting. For example if you have a file called *bigone.dat* that is 10MB and you break it down in chunks of 1MB, you will end up with 10 files named *bigone.dat.01*, *bigone.dat.02*, ... *bigone.dat.10*.

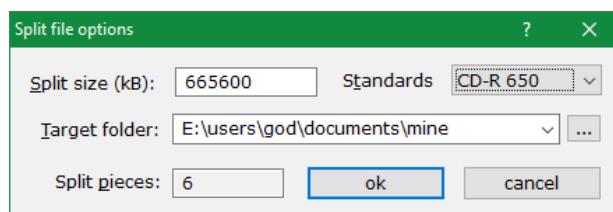




Figure 30. The file split dialog

This will allow you to place the file segments on floppy disks and do the transfer to another computer. There you just need to make sure to place all fragments in a single folder, **select** them all after *sorting by name* (so that they'll be in order), and use **Actions > Merge** menu to put them back together; the file will be restored keeping its original modification date. You can now delete the split fragments!

 **Merge** command will let you join any 2+ files without checking whether they were created by a former split command, if they are consecutive, if all the split pieces are joined (anything forgotten?), or any other reasonable check. You do get warned if you try to merge out of sequence pieces, but it is assumed you know what you're doing!




 You can also merge file segments using the DOS **copy** command. For example *copy part1+part2 whole* will merge file parts 1 & 2 to create a file called *whole*. This is not as convenient as the built-in Merge command; it merely demonstrates that you don't need xplorer<sup>2</sup> to put files back together!


## File and folder comments (tags)

Comments are small pieces of text associated with files and folders, 100% controlled by the user and ignored by the filesystem. There are many possible uses for comments:

- Extended information for the contents of a document
- *Keywords* for document classification, aka tags
- Any other arbitrary information as cross-references etc

You can examine file comments adding the stock **Comment [S]** [column](#) in detailed view mode. Comments also appear in [details](#) pane for the focused item in the active pane. They even show in **Details** property page (**File > Properties** menu) along with other properties — but they aren't easy to browse there. Most important of all, xplorer<sup>2</sup> allows you to use comments as [search rules](#), so you can easily check and retrieve files that have one or more keywords.

Name	Comment
 %A	folder comment
 DOGGYATTACHMENT.pdf	quarantined suspect attachment
 binFileOpFlags.png	photo; manual; installer

 Windows property system supports a comment property for files, which over the years changed implementation many times, and still cannot be applied to all file types (e.g. PNG files). xplorer<sup>2</sup> uses the standard comment storing methods if possible, but complements with its own custom comment formats when necessary to ensure that comments can be set for **all** files and folders. The situation is even worse with *Keywords/Tags* property, therefore you are advised not to use keywords column but store your keywords as comments instead.

You set and modify comments for one or more selected files using **Actions > Set comment** menu command, which displays this dialog:

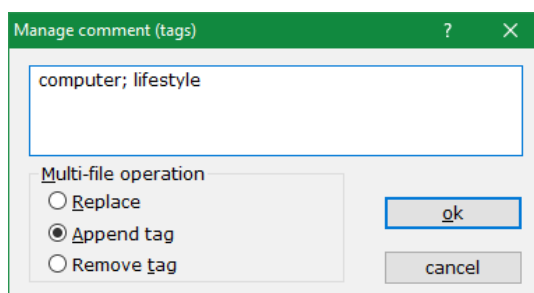



Figure 31. The comments management dialog

Comments can be as long as you like (up to 4096 characters to be exact), that's why you get a big box to type your comments. The dialog shows any prior comments if any exist; if multiple files with comments are selected, you see the comment of the *first* one selected.

When changing the comment of a single file **Multi-file operation** group is disabled because its modes aren't needed: you just see the entire comment and you get to change


or delete it as necessary. When many files are involved you have 3 options:

- **Replace** any old comments with the new one you type (all files will end up having the same comment)  
If you leave the comment field empty, then all comments will be **removed**.
- **Append** one or more "tags" (really whatever you type is added to the comment of each file)
- **Remove** a tag from the existing comments



Simulating tags (keywords) via comments is entirely up to you. You can add single keywords to describe the content. You can use spaces as "separators" or split keywords using ; symbols or whatever. As far as xplorer<sup>2</sup> is concerned, it all is a big pile of letters (it won't mind if you tack on the same keyword twice in a file comment) — and you get to make sense of the information.

When you are happy with the comments and the operation, click **ok** button to set, modify or clear the comments of the selected files. When the operation is over you will see a message on the statusbar "*Operated successfully on xxx items*", which should be equal to the number of selected files — unless there was some problem.



For most kind of files comments are stored in alternate data streams (ADS, see below). As such there are some **limitations** to keep in mind:

- You can easily set comments for all folders, but *files* must be on **NTFS**-formatted partitions. A notable exception is *compound files* (e.g. DOC,XLS and all Microsoft Office documents), and special formats like JPG that store comments in the main file stream.
- You will *lose* comments if you copy files from NTFS to an older filesystem e.g. to a flash drive (FAT32). The workaround is to use the command **Actions > ADS > Bundle to go** menu that preserves all ADS including comments (then use **Actions > ADS > Unpack bundle** at the destination to restore ADS). Compound files don't have this risk though, so they are safe to transfer anywhere without bundling.
- Likewise comments are at risk if you **compress** files with comments, unless you tick the option to include ADS into the compressed archive — please consult your compression tool documentation for more information.

xplorer<sup>2</sup> has limited support for `DESCRIPT.ION` comments but it is a half measure at best. It can read such comments but will not write back into this special file.

### Alternate data streams

File comments must be stored in the file somewhere, and be part of it, but without affecting the "real" content (e.g. the text document). Some file formats like JPG have provisions to save extra information in the main body (EXIF information), but for most plain file types like TXT we must store comments in a way that they won't interfere with the content.

File comments rely on an advanced NTFS feature called *alternate data streams* (ADS). One way to view ADS is like separate *chapters* within a file. What you usually regard as file is the main stream where your data is held, e.g. your text. With NTFS you can associate other streams with a file or folder. So `FILE.TXT` can carry a stream called `OTHER` and it will be addressed as `FILE.TXT:OTHER` using a ":" to separate the stream name. When you move the file around, all these alternate streams are silently carried along.

The standard stream for comments is called **SummaryInformation**, but unfortunately the property system makes a mess of it for most file types, so xplorer<sup>2</sup> uses its own stream called **x2\_AFSComment**. This makes things consistent within xplorer<sup>2</sup> but obviously other programs like windows explorer will not be able to read such special comments.


xplorer<sup>2</sup> has a couple of commands that deal with ADS, in **Actions > ADS** submenu. **View streams** command shows a summary of file streams as the pic to the right, including the first few bytes of each stream (including byte size information). **Split streams** command extracts any *named* streams from a file and saves them in separate file(s), for your perusal.

Alternate data stream info

TYPE	SIZE	NAME	CONTENT
-----			
C:\Users\god\Desktop\Clipboard01.gif			
3	164	(empty)	01 00 04 80 14 00 00 00 30 00 00 00 00 00 00 00 .....
1	5,643	(empty)	47 49 46 38 39 61 B8 01 94 00 F7 00 00 01 01 01 GIF89a.....
4	8	x2_AFScomment:\$DATA	6B 00 77 00 6C 00 6F 00 k.w.l.o.
7	64	(empty)	57 18 9D 9D C7 86 EE 11 83 AB 92 02 37 2C 0A 50 W.....7..P

To **clear** any ADS and leave only the main file content stream, you can copy a file to a flash drive and back, or compress it in a simple ZIP archive then extract it — all extra streams will be washed out!

Another use of ADS is the **Zone.Identifier** information your browser attaches to files you download from internet, which cause various "unsafe" warnings. If you trust the downloaded file, press <ALT+ENTER> to see its properties and tick **Unblock** box in general property page.



There are a couple of stock columns that show ADS information:

- **Streams** shows the number of ADS attached to a file (=1 if there are no other streams)
- **Stream names** lists any named streams present in the file.
- **Size on disk** includes all stream sizes and the main file size — watch out for big differences among this and the regular size property.

### Bulk change of file properties

There's another way to change comments or any other text (or numeric) file property, in bulk for all selected files.

**Actions > File properties** menu lets you change *editable* text file properties like Author, Title, even tags (keywords) and comments. The dialog is identical to that used for [mass renaming](#), only you select the file property to change using the **File attribute** selector box near the top of the dialog. Using a GUI similar to mass renaming you can either set, clear or modify through search and replace the existing file properties. This command is mainly for **text** properties, but it can also be applied to numeric ones, like track number # — just make sure your templates translate to numbers, e.g. using automatic counters \$01.

File attribute:

Comment [S]

Comment [S]

Keywords [S]


Color tag [S]

Contributing artists

Album

Year

The procedure was explained in detail so it won't be repeated here. Just select the property you want to change using the drop-down list, and choose the mode, set your property manipulation template, **preview** the changes, and off you go! Some dialog controls that are only valid for renaming like *Preserve extensions* will be disabled in this function.



If you want to change properties of many files simultaneously, take care to select only **one** file type (e.g. music files), and ensure that it supports the property you want to change. Attempting to set e.g. **Author** information to TXT files will just result in error messages.

## Examine and maintain your filesystem

Daily heavy-duty work on a PC will invariably result in a system that is suboptimal at best and unstable at worst. xplorer<sup>2</sup> has a number of advanced commands that help you restore order and reclaim wasted hard disk space.

### Comparing & synchronizing folders ▶ PLAY

Many times you have two or more folders that have the same contents and you want to keep them *synchronized*, i.e. make sure all "versions" stay updated whenever you make changes to one of them. The typical scenario is a daily **backup** of your current work documents, from your PC onto a safe removable drive, so if a mishap strikes your PC you can retrieve your work with minimum losses from the backup location.

Folder synchronization needs to track all changes between 2 folders, either on a single level or deeply including all the subfolders. Possible changes include:

- Creation of new files and folders
- Modifications of existing files
- Deletion of files and folders

When one folder is the backup of the other, then synchronization runs only in one direction (from the working folder to the backup location), but there could be cases where both folders are "live" and it takes more juggling to keep them synchronized, considering modifications at both sides.

That's when xplorer<sup>2</sup> and its dual pane layout comes in handy. It is ideal for comparing the contents of any two folders, and finding items that are missing from either one, or have been changed in any way. Once you identify the differences, you can decide how to deal with them — e.g. copy the new and modified files manually to equalize the folder contents.

xplorer<sup>2</sup> doesn't have a real synchronization function, it merely **compares** folders and marks the differences. The misnamed menu command **Mark > Synchronize panes** compares the contents of 2 folders, and tries to find file names missing left and right. When the same file exists both sides, it checks the *modification date* to find which version is "newer" and marks it as such. Here is an example comparison:

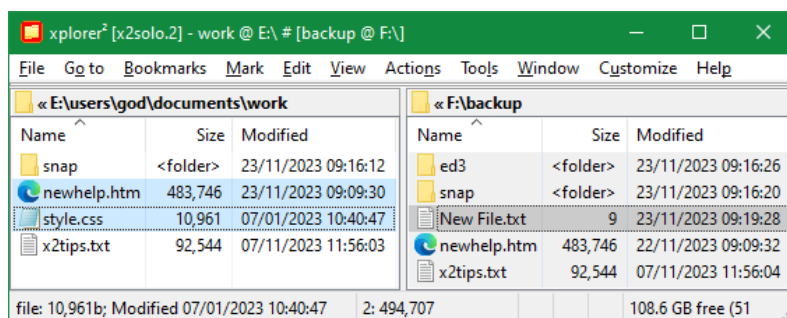


Figure 32. Results of a sample folder comparison

When you press <F9> key xplorer<sup>2</sup> compares the folder contents and marks (selects) **unique** files like `STYLE.CSS` that exists only in the left folder, and **newer** versions of files that exist both sides (compare the modification dates of `NEWHELP.HTM`, the newer version on the left is selected and the one on the right isn't). File `X2TIPS.TXT` has **identical** dates left and right so it isn't selected at all. To really synchronize the 2 folder contents, one must copy only selected files from one pane to the other (this must be done *twice*, once in each direction — observe the unique file `NEW FILE.TXT` on the backup folder to the right).

Once you have the differences selected, you may need to compare the file versions by **content** and see what the modifications are all about. You can use the [preview](#) capabilities in xplorer<sup>2</sup> or external tools that compare files side by side. For plain text files use **WinDiff** or similar tool that highlights the differences — that's how I spend most of my life doing <g>. It is very easy to invoke windiff to compare the focused item in the active pane with its namesake in the inactive with a [user command](#) like: `WINDIFF "$N" "$I"`.

Notice how the plain <F9> "synchronize" command didn't mark any **folder** differences. Folder modification dates aren't very reliable to track changes in deeper levels. We will return on this important matter later.

For more control on what gets marked (selected) after the comparison, use **Mark > Sync wizard** command. The dialog indicates that comparison has two facets; first you determine *how* you want to compare items and then *which* items to mark. The large number of combinations allows you to do all sorts of checks to suit your synchronization needs.

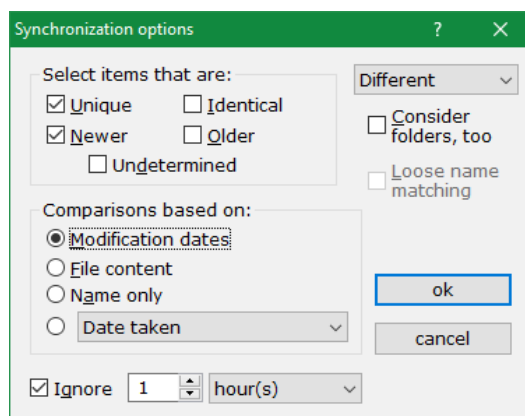


Figure 33. Folder comparison dialog


The comparison logic first tries to match items left and right using the filename as a guide. If a file in one folder doesn't have a matching namesake in the other folder, it is immediately considered **unique** (cf. the checkboxes in figure 33). If a file exists in both folders, the two copies are compared for differences, using the mechanism you specify:

- **Modification date.** This is the fastest way, when the date each file was last changed is used to determine if the two files are **identical**, and if not, which one is **newer** (the other one will then be **older**).
- **File content.** When dates are unreliable for one reason or other, you can check the actual data in the two files. This is a much slower but infallible method to compare for identity or otherwise. Note that when differences are found, there's no way to tell which file is "newer" so xplorer<sup>2</sup> arbitrarily considers the version in the *active* pane as newer and the one in the inactive as older. If in doubt you should open the files and manually inspect the content.
- **Name only.** This quick-n-dirty scheme will just find items that are missing from either folder. Files that exist in both folders are considered "identical" without further




checks for dates or contents.

- **Custom property.** Any piece of file metadata ([column](#)) can be used, typically some date other than last modified. So a file can be newer or identical in terms of date created or date picture taken, depending on the column you select. If you compare items on a text property (not a date) then the one in the active pane is considered "newer" in case of differences.

 When comparisons are based on date modified (or any other date) two items are considered different if their dates are more than +/- 2 seconds apart. You can use the **Ignore** tickbox at the bottom of the synchronization dialog to specify a different interval to ignore. E.g. if a change in daylight savings DST resulted in artificial 1-hour differences, you can set it to ignore this 1 hour — files differing by up to 1 hour will be considered identical as long as their size matches. You can use other time units (up to years) with the drop-down list control.


Given the wealth of xplorer<sup>2</sup> browsing options and comparison parameters, there are many ways comparisons can become "abnormal". Here are a few things to keep in mind:

- **Partial comparisons.** All synchronization commands compare files that are visible — anything hidden by a [visual filter](#) will be skipped. Sometimes you can use this to your advantage to speed things up, e.g. hide all irrelevant files before you do a slow comparison by *content*.
- **Undetermined items.** There are various situations where the relationship between compared items cannot be reliably established. In that case both items remain unselected and a red question mark overlay  drawn over their icons. Possibilities include items with same dates but different sizes, missing custom properties, content read errors etc.
- **Virtual folders.** Comparisons are meant for regular filesystem folders, but many [virtual](#) folders (zipfolders, phones) also have valid date information that can be used for comparing special items with normal folders.

The relationship between files is fixed by their properties and the comparison scheme you select. The second phase is to decide which files you want to **mark**, ear-tag them for further processing. There are checkboxes for all item states and you can check as many of them as you wish - although not all combinations make sense, e.g. selecting both "older" and "newer" files. The drop-down menu on the top-right lists some common scenarios for your convenience (modes for marking different or **identical** items). The plain synchronize command <F9> is equivalent to ticking **Unique**, **Newer** and **Modification dates** options


Ticking **Undetermined** option will mark all those items that were not successfully compared (those with the question marks mentioned above). Tick **Consider folders** option to include folder items in whatever comparison method the files go through — but only in terms of name and date properties, their subfolders aren't considered. For this and **loose name matching** option see the [deep sync](#) section.

When all dialog settings are set to your liking, click **ok** button to start the comparison. Usually this is done very quickly unless you are checking custom slow properties or file contents. A progress bar on the status bar will keep you informed of the comparison progress. If it takes too long you can hold down <ESC> key to abort the operation.

 When you tick **File content** option, files are compared byte by byte regardless of surface date attributes. Imagine comparing 2 *identical* video files of a few GB each, the comparison would have to reach all the way to the end, and take ages! That's why xplorer<sup>2</sup> cheats a little and compares huge files **statistically** rather than exhaustively. The [advanced option](#) "Statistical comparison size" defines what a "huge" file is (default anything over 20MB). It checks **all** the bytes below the huge limit (from the beginning of the file), plus 10% at random above it. If the files are identical, then you get the same verdict at a fraction of the time. Arguably there is a small uncertainty involved, as 2 files may be different in places near the end where xplorer<sup>2</sup> didn't examine, and they are deemed "identical" by mistake. However the advantages far outweigh the small risk. If you want to be more certain, increase the cutoff limit to its maximum value of 250MB.

When xplorer<sup>2</sup> finishes comparing you get the verdict either that "The contents of the two folders seem identical" or if any differences are spotted items are **selected** for further action. For example if you were scanning for differences, you can proceed by [copying](#) the marked files to the "passive" folder, ensuring that the files in both folders are the same, both in number and most recent content. Note that there may be unique and newer items in *both* folders — obviously not the same items — that need to be carried across.

#### Further reading

• copy only newer photos from phone or digital camera [demo video](#) 

### Deep synchronization

Sometimes you may need to synchronize folders including their subfolders, as in the situation to the right. If the hierarchy is shallow (few folders) you can turn on **mirror browsing** (**Go to** menu) and let xplorer<sup>2</sup> keep the inactive pane in the matching folder, as you browse folders in the active pane. As an example, starting from folders A:\ and c:\documents\kenny\help in the active and inactive panes respectively, if you browse into A:\help the inactive pane will automatically follow to read c:\documents\kenny\help. This mirroring will continue in both down and up direction, as long as folders with matching names exist.



When many subfolders are involved, mirror browsing and comparing folders one by one is cumbersome. An easier way to compare two folder hierarchies is to browse the required root folders in a normal dual-pane window and pick **Tools > Compare subfolders**. This command will open a dual scrap window, [flatten](#) each hierarchy in a separate pane and will also launch the synchronization command. If grouping is supported, items will be [grouped](#) by folder helping you to inspect the comparison results. This way you will reveal all the changes that need to be made acting on *all* subfolders simultaneously.

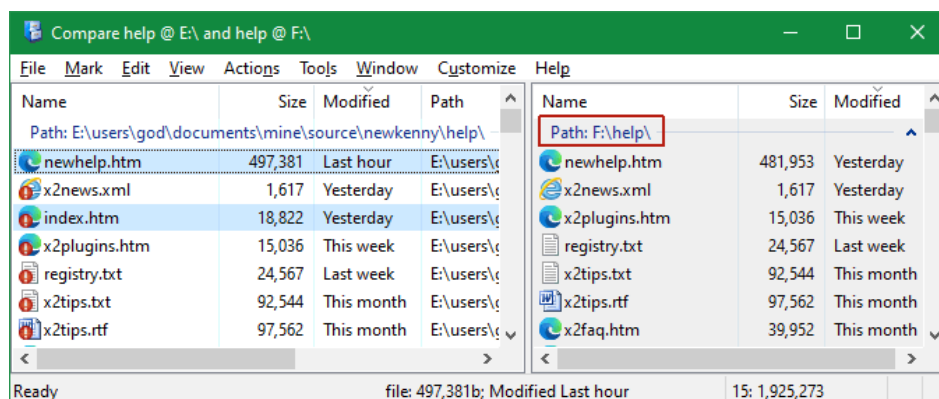




Figure 34. Comparing folder hierarchies


 The normal mode of deep comparison involves only **files** and no folders — the latter are involved indirectly through the files they contain. Sometimes you want to get an overview of the folders involved, find out if many are missing from one side or the other. That's why when you invoke **Compare subfolders** command it asks you: *Do you want to compare only folders, ignoring files?*

Arguably, inspecting the results of a deep folder comparison isn't the easiest thing to do. xplorer<sup>2</sup> turns on **Mirror scrolling** mode to help you examine files left and right, but there may be just too many results to juggle. To protect the marked files, you should turn on **Sticky selection** from **Mark** menu. Another handy feature is pressing <ALT+DOWN> keys to jump to the next selected item, in case they are far apart. Or to focus just on the marked items use **View > Visual filter > Selected only** menu and concentrate on the differences.

Once files are marked, use the special **Edit > Sync-o-paste** command available in scrap containers to copy all the files in their matching *folders*. So items selected in A:\ will end up in c:\documents\kenny, those in A:\help will end up in c:\documents\kenny\help and so on. Note that Sync-o-paste acts on the selected items and doesn't require a prior **Edit > Copy** command.

 Scrap containers may host items from arbitrary locations. The deep synchronization logic tries to match *folders* first, and then applies the usual content comparison




 procedure for each folder pair. Sometimes it won't be possible to match all folders left & right, when the hierarchies are completely different. In such cases there will be a lot of items left in an **undetermined** state, and a little red question mark will appear overlaid on their icon. If you want to compare items that don't belong to similar folder structures, you can try ticking **Loose name matching** comparison option in the dialog ([figure 33](#)); this will compare items matching just names — ignoring folders — but you won't be able to use the sync-o-paste command on the results. Multiple same-name items are matched in the order they are listed in each pane.

Another drawback is that when a folder sub-structure is entirely missing from one hierarchy, all contained items are left in an undetermined state, since their matching folder isn't available.

## Backup synchronization

If you find this form of deep synchronization with scrap containers a bit of a headache, you can try an alternative that is based on [robust copy](#) <F5> command. Selecting **Overwrite if newer else skip** [transfer option](#) will result in copying only files that are new or changed, just as if you did a synchronization based on modification dates — and it doesn't need scrap containers or any prior synchronization commands! This will work best if you have all the changes in one folder hierarchy and you want to mirror them to some other "backup" location like a removable drive.

 Always use a removable drive for safe backups. There is no point saving in the same hard drive as you keep your work, if something goes wrong with your PC you will lose your work *and* the "backup". For resilience, keep the backup storage separate, ideally in a different room of your house — or at your mum's house even better. Squirrels spread their bets to be safe <g> As for **cloud** backups, I have trust issues handing out my most prized digital possessions to some private cloud entity. DIY if you can!

Here is a summary of a backup operation:

1. In a dual-pane (normal, *not* scrap) xplorer<sup>2</sup> window, browse your work folder (e.g. MyDocuments) in the left pane, and the backup location on the right (or bottom if you have your panes set up/down).
2. Click on the left pane to activate the work folder, where you want to copy files *from*; the inactive pane will be the *destination* for the backup.
3. Select all items with <CTRL+A>
4. Use the robust **Edit > Copy** to dialog; select **backup** option (it is available in the options drop-down list, see [figure 22](#)). If you can't find it there, use the **Options** button to tick "Overwrite if newer else skip" box.
5. Click **ok** to start the backup copy.

The above steps 3-5 can be simplified if you use **Tools > Backup copy** menu command. You only need to browse the correct folders in that case (steps 1-2) and let the backup command handle the rest automatically.

Except for the very first time you create a backup (where everything must be copied for the initial snapshot), in subsequent backup operations only newly created and modified files will be copied, which is exactly what you need, and you save lots of time by skipping all the files that remained the same. As robust copy is recursive, you get all the folder hierarchy synchronized — in one direction. So you have achieved efficient, deep level synchronization without bothering with comparing and inspecting!

You can manage your documents so that they all fall under a single root folder like MyDocuments — organized in various subfolders of course. This kind of hierarchy makes backups very easy as you only do a single backup for *all* your documents. If on the other hand your work folders happen to be scattered in different places, say some in C: drive and some in E:, you can use [folder junctions](#) to create a single root "master" folder somewhere, and drop in there junctions to all the various work subfolders. Thus you have reorganized your important storage without actually moving anything, and backups can be done in a single step. For this trick to work you must enable the [advanced option](#) "Treat folder junctions as normal folders..." and all xplorer<sup>2</sup> recursive commands will follow junctions automatically.

### How often do you backup?

According to *Sod's law*, the chances of losing your digital work through mishaps grow proportionately with its importance. Therefore creating a safe backup copy once a day isn't too far fetched. xplorer<sup>2</sup> makes it quick and easy so you can even consider more frequent backups.

## Checking file transformations

The folder synchronization command just described checks for modifications of different *versions* of the same file. A variation on this theme is checking the state of file transformations, when the compared filenames are slightly different.

If you are involved in programming, then you will recognise this situation as *compilation* of source code to object files. So starting from a C++ file `xplorer2.cpp` the compiler churns out the machine translation in a new file called `xplorer2.obj` - notice the change of extension from **.cpp** to **.obj**. **Mark > Check build** command checks the states of source files comparing them to their "transformed" versions.

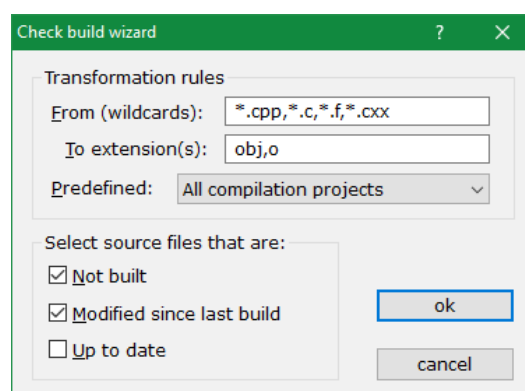



Figure 35. "Make" check dialog

The dialog shown in the above figure accepts as input the transformation *rule*, which observes the source and target filename extensions. It can be seen that multiple comma-separated extensions can be used to check for all compilation projects in one stroke. Also note that a generic [wildcard](#) is accepted for the **From** field whereas the **To** field must be just the target extension(s), *without* the "." (e.g. type **obj** instead of **.obj**).

 You don't need to supply more than 1 extension if you target only a particular transformation e.g. from \*.PNG to GIF. The **Predefined** drop down list has a few sample transformations that are programming oriented. You can add your own extensions to the predefined list through registry editing.

The idea with this command is to browse the "source" items in the active pane, and the "transformed" equivalents in the inactive (both could be in the *same* folder), then tick the boxes to locate items that either need transformation or they are **Up to date**. Click **ok** button and **Check build** command will then examine names and their filetimes and select *source* items in the active pane according to the checkboxes you ticked.

Once source and target files are matched their modification dates are examined to figure out which is newer. If the source is newer, it must have been **modified since last build** (cf. [figure 35](#) checkboxes) and it requires re-compilation. If the target is newer then it is **up to date**, whereas if it doesn't exist at all the source is **not built**. You can check as many boxes as you want to tell xplorer<sup>2</sup> which *source* files to select. Let's take an example.

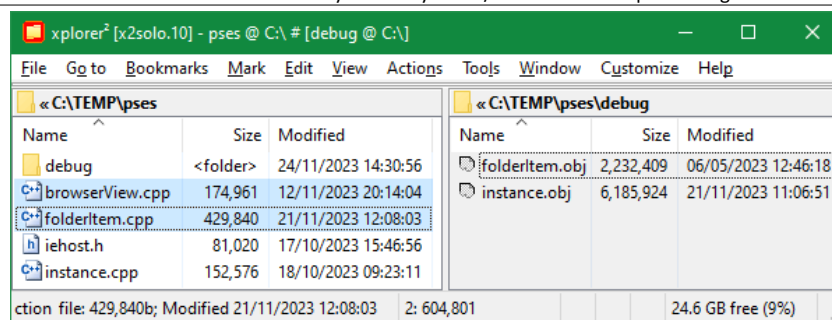



Figure 36. Sample project check

Using the options in figure 35 on a pair of hypothetical source and target folders, xplorer<sup>2</sup> marks all source files that require compilation. The file `folderItem.cpp` was modified after it was last built (check the date of its target `folderItem.obj`) so it is marked. `browserView.cpp` hasn't been compiled at all (no `browserView.obj` exists in the inactive pane), so it is "not built" and hence marked. `instance.cpp` on the other hand is up to date, and finally `IEHost.h` is completely irrelevant for this transformation rule so it is left unselected. Note that *nothing* is selected in the inactive pane to the right; this command is checking one way only.

Like the folder synchronization command, **Mark > Check build** merely ear-marks files that require building without actually taking any further action. It is up to the user to decide what to do with the selection afterwards, e.g. generate a [script file](#) that executes the compiler for each source file.

 This oddball command doesn't make claims of being a substitute for **make** type tools. For instance it cannot check for cross-dependencies from modified header files etc. Still it can be useful for simpler scenarios like **MatLab** projects compilation, or batch converting image files from one format to another.

## Visualize disk space usage

According to the second law of computers, no matter how big your hard disk, eventually it will fill up with movies, videos, pictures and what have you. Do you need all that stuff? Is space wasted in temporary and duplicate files? xplorer<sup>2</sup> has some tools to let you *focus* your disk cleanup effort at the biggest offenders, files and folders that take up the most space.

**Tools > Folder statistics** menu command creates a detailed analysis of the active folder and its subfolders. If you want to examine your entire hard disk, first browse the root `C:\` folder then invoke the statistics command — naturally the bigger the hierarchy, the longer it will take to analyze. It will calculate details for each subfolder like number of files and size used, and will do so recursively for all the contained folders, and present the results as a tree:

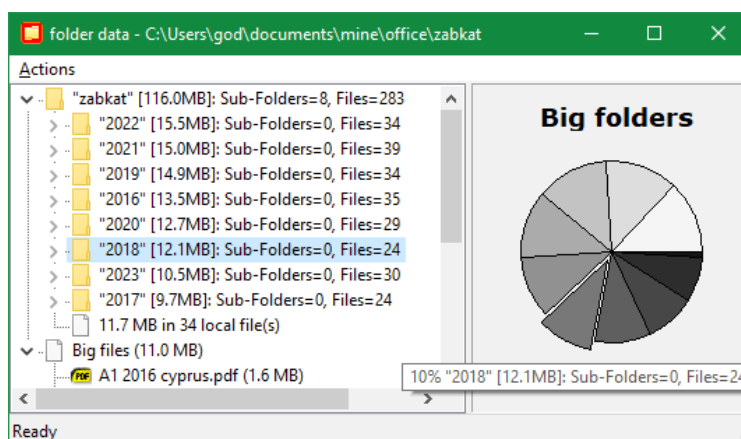


Figure 37. Folder data window

The largest folders appear first. To aid visualization of space usage, a little pie chart is shown too, popping out a pizza slice for each folder you select in the tree. Drag the splitter bar with the mouse to make the pie bigger or smaller. You can also use the tooltips over the pie pieces to discover the folder names.



The second main tree branch shows the 10 biggest files in the hierarchy examined, arranged by size. Examine the **big files** branch to see whether you don't need some of them and you can delete them to save space. Select a file and press `<ENTER>` key to browse its containing folder.


The tree is for information only. You can examine and open branches, but you cannot do much file management. If you want e.g. to delete a large unwanted folder, use **Actions > Browse** menu (or double-click on the folder) to open it in a regular xplorer<sup>2</sup> window. The tree won't autorefresh either, so if you delete a folder branch, use **Refresh** command to update the tree information.

The single menu bar item called **Actions** holds a few commands, the most useful of which perhaps is **Copy**, which copies the entire selected tree branch as text, that can be subsequently pasted in a text editor for printing or whatnot. Use **Find** and **Find next** to search for folder names (this will open tree branches automatically). You can also switch the sort mode from size to sort by **Name** — although the former arrangement is more useful in this context.

Another useful menu command is **Empty folders** which detects folders that are empty (they have no file contents), and puts them in a [scrap container](#) for further action (e.g. select all of them and delete them). Empty folders won't take up much space but some people want to get rid of them regardless. This command is smart enough to discover the *topmost* empty folder in a local hierarchy, and ignore its empty subfolders (if any) — that will be deleted with their parent folder anyway.

Once you discover the biggest size offenders, clean up your space deleting junk files, e.g. [duplicates](#), [similar pictures](#) etc.

	Browse	Enter
	Sort by name	
<input checked="" type="checkbox"/>	Sort by size	
	Copy text	Ctrl+C
	Find...	Ctrl+F
	Find next	F3
	Empty folders	
	Refresh	Ctrl+R
	Close	

 If you analyze your entire `C:` drive you will discover that `C:\WINDOWS` is probably the biggest folder. However you shouldn't delete stuff inside it — you may save space but at the expense of your system stability! Windows folder size isn't as bad as it looks anyway because many "duplicate" DLLs are hard linked so the space usage is less than that reported by folder statistics.  
To cleanup space from `C:\Program Files` uninstall apps and tools you don't use.

## Continuous folder size

Subfolder objects *themselves* do not take much space, that's why in detailed view xplorer<sup>2</sup> prints just `<folder>` in the size column. Obviously the *files* inside a folder take up space, so it is possible to see this information for each folder — in regular xplorer<sup>2</sup> windows that is.

While browsing a folder with subfolders, **Tools > Subfolder size** command will show you the size of each individual subfolder, so you can tell which one takes most space. The size shown is **recursive**, including space occupied by all deeper level subfolders. This command also works in special folders like phones and zipfolders.



Recursively adding up subfolder sizes takes a lot of time and effort. As an optimization, **Subfolder size** command will only calculate sizes for *selected* folders. If there is no folder selected then *all* sizes will be calculated.

Name	Size	Contents
📁 snap2	785,333	118
✅ in	16,536,185	317
📁 phplist	<folder>	63

Enable the stock **Contents** column to see the number of items (subfolders and files) contained in a folder. This information is *not* recursive, it won't tell you about files deep inside sub-subfolders, only direct folder contents.



The contents column is sort of programmable. Instead of calculating *all* contained files and folders it can count through a wildcard (or a comma-separated list of wildcards). Thus you can only count say, \*.MP3 files contained. This tweak is an **advanced option** called "*File count rule*". It can also be the name of a saved hyperfilter — but only fast properties are allowed

If you want to see subfolder sizes all the time, without pressing <CTRL+D> for manual subfolder sizes, activate the **option** "*Calculate subfolder size automatically*". This is a major resource hog, so if you are browsing a top level root folder like C:\ it won't work automatically. Sometimes calculating these sizes takes so long that you may browse a different folder before the recursive addition is complete — definitely not recommended!

xplorer<sup>2</sup> doesn't cache folder sizes, so each time you visit a folder the sizes are calculated again and again. This is totally wasteful, so if you like to see folder sizes and have pity on your system resources, please download a plugin called **folder size**, that works well with xplorer<sup>2</sup>.

#### Further reading

- Remedies for reclaiming critically low hard [disk space](#)
- Delete intermediate [Visual Studio](#) project files to free space

## Detecting duplicate files ▶ PLAY

Everyone is a **digital hoarder** nowadays. We take hundreds of photos with our phones, download music and movies, accumulate work documents, and we want to keep it all stored forever. Over the years many file collections are infested with duplicate or items that just waste disk space. xplorer<sup>2</sup> will help you discover and reclaim this wasted storage space.

Duplicates come in many guises. In the simplest case a file will exist in many carbon copies using the same name. A little bit trickier is when a file exists with many different names but the same data content (renamed). Check duplicates command will help you detect all these kinds of extra files.

Duplicate files could reside anywhere, in your PC's hard drive or in external storage. The first step is therefore to load all the folders you want checked in a scrap container. If you want to check on external hard disks you must have them all connected. Here are some possibilities:

- Browse the top folder you want checked (e.g. MyDocuments) and use **Tools > Check duplicates** menu command. This will load all *files* of the active folder and its subfolders (optionally for the *inactive* dual pane, too) in a scrap window.
- Manually build a [collection](#) of files to check. You can add folders to scan dragging them with the *right* mouse button then using **Browse flat** from the drop menu. Once your collection is complete, use **Tools > Check duplicates** from the scrap window's menu bar.
- A combination of the above methods: use **Check duplicates** command in the main xplorer<sup>2</sup> window to browse flat a hierarchy, temporarily *abort* the scan, then add extra folders e.g. from external devices using right drag in the same scrap window.

#### Have you tried i-DeClone?

i-DeClone (another tool by ZabKat) can find exact and **near-duplicate** files (e.g. the same movie saved in different formats). i-DeClone helps you find more duplicates and is better streamlined for the cleanup task. Help yourself to the [free trial](#)



You should focus on your **personal** documents and ignore duplicates in system folders like %WINDIR% or AppData. There is usually waste in these folders but deleting hidden and system files is unsafe and may destabilize your PC. Checking your entire ThisPC folder for duplicates is possible but will take a long time.

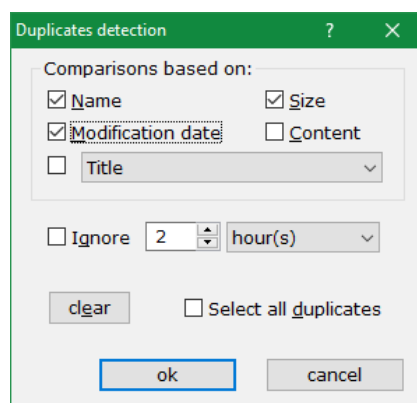


Figure 38. Duplicates' detection parameters dialog

Next you specify the comparison criteria with the above dialog box. The options set in the dialog determine how the investigation is to proceed. You can have any combination of boxes checked (use **clear** button to reset all checkboxes before a fresh search) but the fastest would be to assume that duplicates would be 100% exact copies, that is they will have the same **name**, **modification date** and **size**.

If you are comparing modification dates, there could be phantom differences introduced e.g. by FAT32 partitions, that are only 2-second accurate for stored files. For such cases use **Ignore XXX seconds** box in the dialog. You get to choose how many seconds (or hours!) to ignore when comparing modification dates. Any 2 files that fall within the date tolerance limit are considered to have the same date.

Custom file properties can also be used as criteria for "sameness". Select the property you want using the drop-down box and you can extend the duplicates tester in creative ways (e.g. find all music files by the same artist or owner).




The ultimate proof that 2 files are identical is their **content**. Files may have the same name and size but differ in their contents. This is even more so if you go for **custom** file properties like Author or Title. Therefore the results of the comparison should be taken with a pinch of salt. Don't delete MP3s as "duplicates" only because they belong to the same Artist!

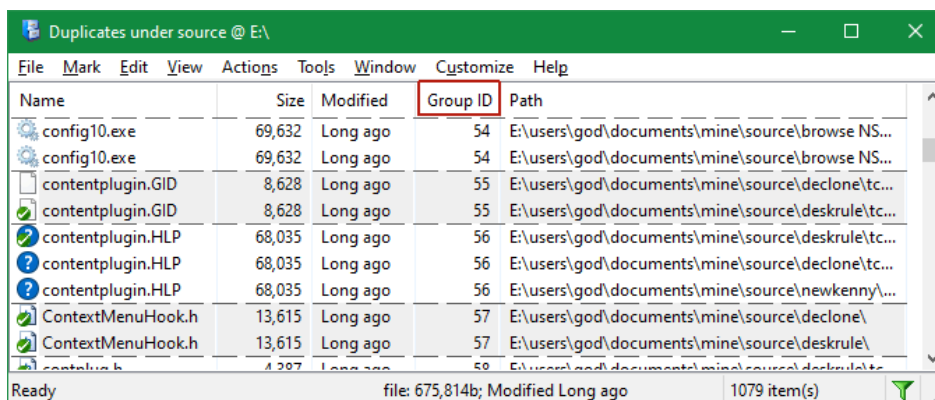
If you suspect you have identical files with different names or/and timestamps, then you can clear all boxes except for **content** (same **size** is then also implied whether you tick the box or not). This is the most robust detection mechanism but at the same time very time consuming, since each file is compared to *each and every* other same-sized file in the container, regardless of their name.

To speed-up content comparisons xplorer<sup>2</sup> uses the same [statistical](#) content trick as in synchronize command, where only a portion of *huge* files is examined instead of each and every byte. Also *small files* are excluded because there may be too many of them and you wouldn't save much space by their elimination. The **advanced option** called "*Ignore small duplicate files*" defines the limit of what is considered "small file" (default 1KB and under)

If you intend to delete duplicate files, please make sure you check **Select all duplicates** box in the dialog *before* you go ahead with the command. xplorer<sup>2</sup> will assume that the **first** item in each group is the "original" and select the *remaining* items for deletion.

When you are all set, click **ok** button to start the duplicate discovery process. Depending on how many items are in the scrap window and your scan options, the comparison may take long time. A progress bar will give you an idea of how much work is left to be done.

 Ultimate xplorer<sup>2</sup> edition has among other things a much quicker [duplicates detection](#) algorithm, that allows you to search your entire hard disk for duplicates and get results in reasonable time.  
There's also a new zabkat tool purpose built for finding duplicate and similar files, called [i-DeClone](#), which is even more capable!



Name	Size	Modified	Group ID	Path
config10.exe	69,632	Long ago	54	E:\users\god\documents\mine\source\browse NS...
config10.exe	69,632	Long ago	54	E:\users\god\documents\mine\source\browse NS...
contentplugin.GID	8,628	Long ago	55	E:\users\god\documents\mine\source\declone\tc...
contentplugin.GID	8,628	Long ago	55	E:\users\god\documents\mine\source\deskrule\tc...
contentplugin.HLP	68,035	Long ago	56	E:\users\god\documents\mine\source\deskrule\tc...
contentplugin.HLP	68,035	Long ago	56	E:\users\god\documents\mine\source\declone\tc...
contentplugin.HLP	68,035	Long ago	56	E:\users\god\documents\mine\source\newkenny\...
ContextMenuHook.h	13,615	Long ago	57	E:\users\god\documents\mine\source\declone\
ContextMenuHook.h	13,615	Long ago	57	E:\users\god\documents\mine\source\deskrule\
contentplug...	4,297	Long ago	58	E:\users\god\documents\mine\source\deskrule\...


**Figure 39.** Sample duplicate file groups

After the command finishes its work, all files that match the user criteria are bunched together in groups with different background color as in the above figure. All unique items are *hidden*. Note that some groups are made of 3 or more duplicate items.

Notice the **Group ID** column, which counts the duplicate group numbers. This is a pseudo-property as it doesn't corresponds to the files, only their relationship — but it is very useful regardless. If you change the sort order examining e.g. file dates or whatever, click on **Group ID** column header to restore the duplicate groups.

Such listings provide preliminary evidence of duplicate files. Depending on your dialog parameters you may need further proof before actually deleting the redundant copies. A good guide is the *checksum* column that you can activate using **View > Select columns** menu. It shows a numeric "summary" of a file's contents. When checksums are different files are 100% different; equal checksums on the other hand do not guarantee identity — although they imply a strong possibility that the files are identical.


To build up your confidence use the [previewer](#) pane to examine file contents. If you are working with images switch on **Detailed thumbs pane style** to see little thumbnails for each item. If you are working with plain text files you can use your trusty Diff tool to check pairs of files: select the 2 files then **Tools > Run command** menu to launch WINDIFF \$A to compare them line by line (\$A [token](#) represents the 2 selected items).

 If you use [hard links](#) they will come up as "duplicates" but most likely you don't want to delete the "copies". Use the **Links** stock column to see whether duplicates in the same group are hard links of each other (Links would be equal to the number of items in the group)

If you want to keep a file group entirely, select its items and press <DEL> key — in scrap windows this merely does a **File > Remove** command, the files aren't deleted. Otherwise to *really* delete duplicate files you must first **select** them. This is easier said than done when you are dealing with thousands of duplicates.


One easy method is ticking **Select all duplicates** in [figure 38](#) before you even start checking. This command can't tell which is the "original" out of the identical files, but it uses the *initial sort order* of the items, leaving the first one in each group unselected and marking all the others. If you want to keep the *older* copy then make sure you sort by **Modification date** — ascending from older to newer — *before* starting the duplicates checker. Thus in each group all items but one will end up selected and ready for deletion.

If this automatic selection doesn't suit you, then you need to examine each group individually and select duplicates manually. As you finish with a group, delete the selected items immediately using **File > Delete** menu — or pressing <Ctrl+Delete> keys. This slow step-wise deletion will spare you any selection mishaps.

 The duplicate results GUI really needed to offer **checkboxes** (like i-DeClone does it) instead of relying on fragile selection states. Use the following safety measures to protect large selections as you scroll and examine duplicate groups:

- Turn on sticky selection mode <ALT+S>
- Periodically save the selection via **Mark > Selection > Store** menu; then you can reapply the selection with the same [submenu](#).

Technically speaking, when duplicate files are shown, the pane is both *filtered* since unique files are hidden from view (notice the green filter icon on the statusbar of figure 39), and *unsorted* since files are displayed in groups regardless of the original sort order. You can cancel this mode using **Tools > Reveal unique** or **View > Show all** menu command.

 The following commands till the end of this section apply to scrap containers only, they don't exist in the main window menu bar. You must open and [populate](#) a scrap window with the files you want to examine.

## Discover similar and bad quality photos

Another source of clutter in your hard disk is your photo collection. It is very easy to snap lots of photos with your mobile phone or camera, but not all of them are worth keeping:

- **Bad quality.** Over- or under-exposed pictures, blurry or shaken, they needn't be preserved for posterity!
- **Near duplicates.** If you took 10 pictures in rapid succession to capture the moment, do you need to keep all of them?

xplorer<sup>2</sup> has commands that let you quickly discover bad and similar pictures, offering you the option to delete them and clean up your disk space. First you can choose the **Blur** file detail (use <ALT+K> keys to choose columns in detailed view mode). For JPG pictures it shows a number, which is big for pictures that are suspect; values greater than 100 indicate bad quality, whereas smaller values indicate smaller defects. Sort your picture collection by this column to bunch the bad pictures together, then inspect and delete what is not worth keeping.

To get rid of similar pictures, [flatten](#) a picture folder or use any other means to fill up a scrap container window with JPG photos to examine. Use **Tools > Find similar photos** menu command (or if using the Ribbon UI see under Duplicates button on the **Home** tab) to group them by similarity. Whereas checking for duplicates will only find files (pictures) that are exactly [identical](#), this command finds pictures that are slightly changed.

### AI quality detection

Both blur and similar photo detection use artificial intelligence SVM algorithms. As such they are good but not perfect. A photo showing only blue sky might appear "blurred" to AI but you should know better. Always preview photos before deleting them

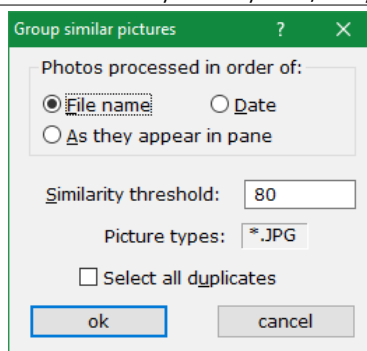


Figure 40. Find similar pictures dialog

The dialog controls how similar pictures are found. To achieve reasonable detection speeds, it examines pictures either by filename or by modification dates, so that it only considers pictures that are likely to resemble each other (shot one after the other). Alternatively you can choose **As they appear in pane** radio button if you want to be in charge of the order photos are examined (e.g. in whatever sort order is in effect). At present only JPG photos are compared, any other file types you add in the scrap container are ignored.

You can tweak the **similarity threshold** up or down, to match pictures that are more closely (or loosely) related. A value around 80% is a good compromise discovering similarities without too many false duplicates — mistakes cannot be ruled out however. Even ChatGPT isn't flawless, never mind lowly SVM detectors.

Click **ok** to start the comparisons. Any similar photos found will be placed in groups, as per the duplicate checker [function](#) (i.e. in bands organized by **Group ID** column) and thumbnail view mode will be automatically enabled to let you compare the similarities discovered. Then just select and delete those you don't need, leaving 1-2 best pictures per group. Select two photos and press <ALT+SHIFT+Q> to preview them side by side in a large popup window (similar to [peek](#) preview but for a pair of photos).

Tick **Select all duplicates** in advance to mark photos for deletion automatically. However this will usually keep the **first** photo (in terms of date taken) as "best" which may not be the case. Always examine the groups yourself. Optionally turn on the **Blur** column to see which photos are technically of poorer quality.

The workflow is similar to duplicate results, you must somehow **select** what gets deleted and be careful to preserve the existing selection. Some groups may be totally rubbish so you can select for deletion *all* photos in such groups. In the end use **File > Delete** menu command to remove the unwanted photos.

## Fuzzy groups

Fuzzy grouping is an interesting (albeit oddball) feature that could be put in various uses. It uses fuzzy matching to group together items that are *similar* but not identical. For example, we can put all our music album folders in a scrap container (use right drag then [Get subfolders](#) from the drop menu), then use **Tools > Fuzzy groups** menu command to group the folders by *name* similarity. Just watch the groups in the following picture, which shows the dialog and the resulting groups simultaneously:

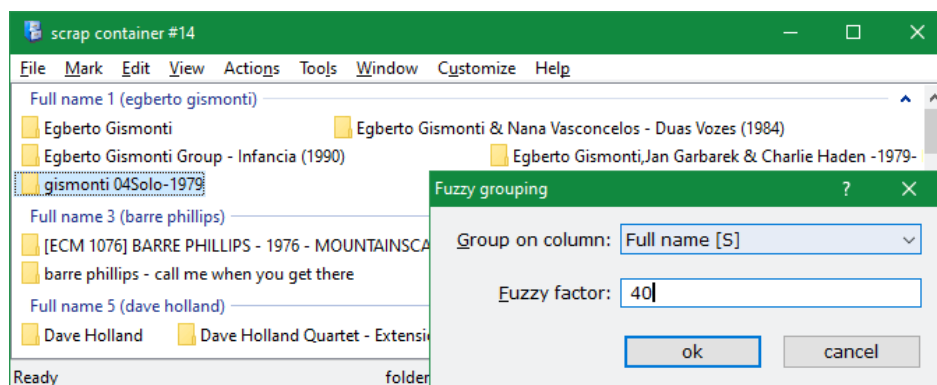


Figure 41. Group similar filenames with fuzzy grouping

As you can see, using a 40% fuzzy factor, the first group contains all folders that contain "Gismonti" in the title, the third group has works by Dave Holland and so on. If you make the factor bigger (say 60), the fuzziness *decreases*, so items have to be more alike to be grouped together. For best results try a few different values until you hit on the desired groupings. You can use any text column for fuzzy grouping, it doesn't have to be by Name only. Just pick what you need from **Group on column** drop-down list. If you pick a number or date property it will be considered as text too — as it appears in the window.

Fuzzy grouping is a special case of [custom grouping](#), only here the command defines the categories automatically using the property and fuzzy factor you supply.

## Even weirder fuzzy groups

If you specify a *negative* number as fuzzy factor, then files will be **clustered** together in terms of mutual distance. The negative number you specify will be the number of clusters (groups) — e.g. **-5** will create 5 groups. Clustering is an AI algorithm that bunches together items that are "close" together, e.g. if their dates are similar. At present you cannot cluster on text columns like Name or Comment, only numeric and date properties can be used for **Group on column**. Any items that don't support the selected property will remain ungrouped in Unspecified category.

If you use a [regular expression](#) (instead of a number) as the fuzzy factor, you can identify **common substrings** in filenames or other properties, and group on them. Let's use an example, consider these files:

```
alpha (2002).mp4
beta (2002) action.mp4
New File 2123
old (2001).mov
alien (2001).mp3
```

Four of these have a year, or 4 digit number, in brackets. If we use the regular expression `\(\\d{4}\\)` that matches this pattern for a fuzzy factor, we will generate these automatic groups:

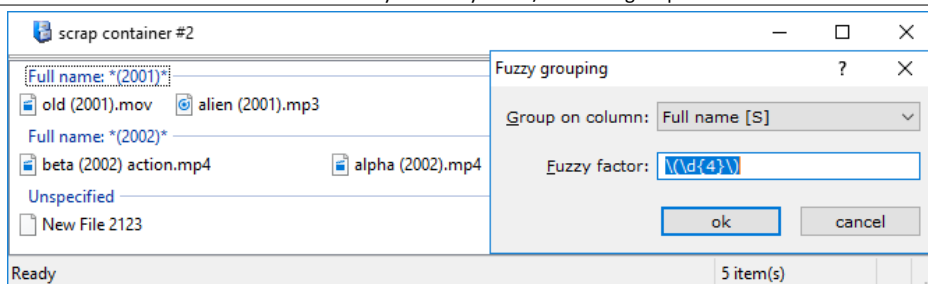


Figure 41a. Group filenames with common parts using regular expression

See how the file that doesn't have a year in brackets ends up in the **unspecified** (non-matching) group.

Here's another use of this command. If you browse `shell:recent` folder of recently accessed documents, you cannot arrange by file type because all files in there are shortcuts; the .LNK extension is masking out the real file type. You can work around this problem putting all items in a scrap window and do a fuzzy group on NAME [S] property (not full name) and this regular expression (that matches the extension past the dot):

`\..*`

You will see that files are grouped by their original extension!

## Convert groups to folders

Regular (sort column) or custom [groups](#) (even fuzzy grouping) in scrap container collections can be converted into actual folders using **Tools > Extract groups** menu command.

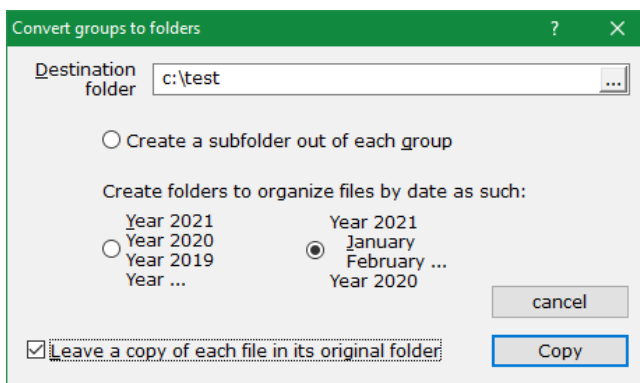


Figure 42. Convert groups to folders dialog

The dialog has 2 modes, or two ways to generate folders, one for real groups (tick **Create a subfolder out of each group**), and one for *implied* groups by date — the latter has 2 sub-modes, see below.

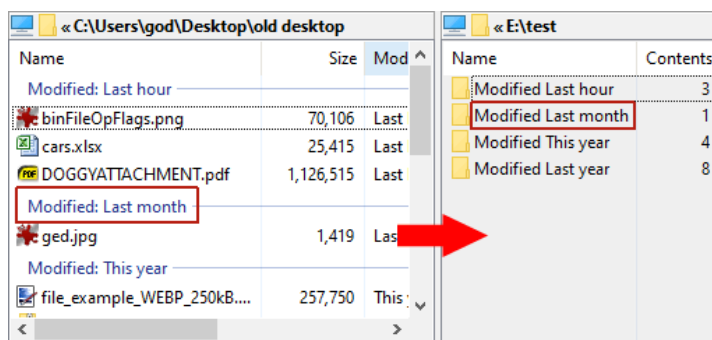
Say you used **View > Arrange by > Show in groups** menu to show items alphabetically grouped using their first letter, you can then use extract groups command to create folders that contain the items of each group. You choose a top **destination folder** and xplorer<sup>2</sup> creates subfolders under it, one per group (the folder names match the group titles you see). You can either copy or move the original files into these subfolders, depending on the checked state of **Leave a copy...** box.

Click **Copy** (or Move) button to start organizing files into group folders. Simple explorer copy routines are used for the transfer. To see the newly created folders you must browse the destination folder you specified (e.g. C:\TEST in the above example).

For example if we had a **scrap** window showing groups by date (see the picture to the right) and used this dialog to split the groups, it would create one folder for each of the date categories — folder name matching group name — and place all the files of said group in their respective folder.

Unlike other xplorer<sup>2</sup> commands, all the visible items are sent to group subfolders, not only those selected. A few other **irregularities** to note with this command:

- Only regular (filesystem) files are handled, and *not* any folders
- Files in deep paths (>=256 characters long) are ignored
- The **unspecified** group (if it exists) isn't exported



This command can also organize items by **date**, whether you have groups or not. This is useful e.g. if you have a truckload of photos in a scrap and you want to automatically move them in subfolders according to their modification date. Use the two date radio buttons to choose how to have them organized, either one folder **per year**, or you can have items organized by year **and month** (i.e. each year subfolder has 12 subfolders corresponding to months). Folders are only created for items that need them; there may be *gaps* in the sequence if files for a yearly period (or month) are missing.

Usually **last modified** date is used for automatic folder creation, but you can pick any other date you want (e.g. **picture date**) by placing it *before* the date modified in the details. To reorganize column order either use ALT+K command or drag-drop using the column header.

You can apply additional order to your photos using the export groups command in stages. For example, if you want to organize your photos by **camera model** as well as by date, so that you have one folder per camera model, and inside that you organize by year/month, you can do it as such:

1. Put all your photos in a scrap container e.g. using search or [browse flat](#) menu command
2. Add the [file detail](#) you want to organize by, e.g. camera model
3. Sort by this column (click on its header) and turn on groups (**View > Arrange by > Show in groups** menu)
4. Convert the current grouping into folders. Now you have one subfolder per camera model.
5. Enter each of these subfolders and reuse **Export groups** command, now using date information to create the sub-sub-folders. For this last step make sure you **move** items (untick "leave a copy..." box), so that your photos are only stored once.

Step 5 is a bit fidgety, because it must be repeated for each camera model folder created in step 4. For destination folder use the same path as the camera model folder created in step 4.

This procedure could be automated using [macros](#)



## Search for files and folders ▶ PLAY

xplorer<sup>2</sup> unleashes the full potential of file searches since you can use *any and all* pieces of file information as parameters. You can thus home in to the files you are after with unparalleled precision avoiding irrelevant hits. You can search a few hand-picked folders — even within zip archives, or whole networks, and even *refine* search results.

You initiate a search using **Tools > Find files** command. The ensuing dialog (figure 43 below) is relatively simple considering the amount of information it conveys. Broadly speaking it is divided in two sections, the search *context* [1] that specifies *where* to search, and the *hyper-filter* [2] that controls *what* to search for, the parameters of the files that will be considered positive hits.

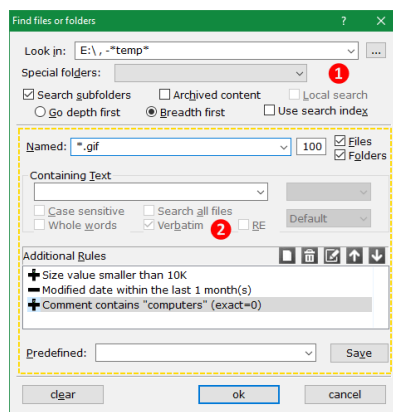



Figure 43. Find dialog: scope and filter selection


First we concentrate on the top portion [1] where we define which folders and how to search in them. By default the search context to **Look in** is the folder you were browsing when you issued the <Ctrl+F> find command. This is the **root** search folder and everything under it will be searched, all the way to the deepest subfolder. You can also specify broader contexts using the **Special folders** drop down box. This includes popular search locations like Documents, local hard drives, This PC that includes anything attached to your PC and any mapped network drives etc. Selecting a special folder as root for the search overrides **Look in** field.


 You can search *multiple* locations too: the **Look in** box accepts multiple comma-separated folders; e.g. typing C:\ , D:\TEMP will instruct xplorer<sup>2</sup> to search both these folders. You can also *exclude* folders using a leading minus sign, e.g. C:\ , -\*temp\* will search all C:\ except any folders with "temp" in their name. The excluded portion must be defined as a wildcard.

Usually you will want to search all the subfolders below those specified in the **Look in** field, as long as **Search subfolders** is checked. You can also control the traversal method: **Depth first** will favor files in deep subfolders whereas **Breadth first** will scan the folder tree in layers. Both methods will scan the entire searched folder hierarchy eventually.

If **Archived content** is checked, xplorer<sup>2</sup> will step into zipfolders, FTP sites and pretty much any other unusual folder that comes in its way. You can use all stock file properties and even search for contained text just like in regular folders. The down side is that searches become much slower, so only tick this box if you know your target file is within an archive.

By checking **Local search**, you instruct xplorer<sup>2</sup> to search within the files already contained in the scrap window, possibly inserted by an earlier find command. This essentially *refines* the search. A similar result can be achieved using **Mark > Matching a rule** command in a scrap container window.

 **Use search index** is usually ticked for ultimate edition and makes searches much **faster**, especially if you search for contained text. This will be elaborated in a following section. The only case you want to *untick* this time saving option is when you search for symbols and special characters that windows search does not index.

 The search scope usually includes all folders under the root of the search, but there are some exclusions as follows:

- Folders with **Hidden** attribute will be skipped if "Show hidden files and folders" is [disabled](#).
- Folder junctions that fork out to remote parts of the filesystem are skipped unless the [advanced option](#) "Treat folder junctions as normal folders..." is ticked
- You can exclude certain folders by path or wildcard using "Find blacklist" advanced option. These folders are also excluded from **Browse flat** command.
- Manually add exclusion wildcards in **Look in** field via the leading minus (-) notation mentioned above.
- Add a boolean *Not* rule on [Path column](#); Folders excluded thus will be entered (so it's not a very efficient method), but no files will be found in them


### Search criteria

After you setup *where* to search it's time to define what to search *for*. You may recognize the framed dialog [section \[2\]](#) as a familiar [hyper-filter](#) that defines the properties of the files and folders we're trying to find. As we've already presented these filters in detail, we'll just brush them quickly here.

When it comes to defining the search parameters, you are really spoilt for choice. The search *hyper-filters* defined in the dialog comprise:

- A **wildcard** (**Named** field) to search for particular filenames or types (e.g. \*.txt, \*.dat). The drop-down portion includes some common document types (e.g. audio, video etc) for your convenience. The little number box next to the name can do fuzzy searches or even regular expressions matches on names.
- Checkboxes for limiting the search to **Files** and/or **Folders** (or both).
- Contained text criteria, when you are after files that must contain some string. All the functionality of the [find text in files](#) command is available embedded in the find dialog. Regular expressions can be used too, checking the **RE** box.
- **Additional rules** based on information available in file columns. You can add rules that match file dates, byte sizes or any other [file property](#) that's available — all of them can be used for searching!

You can define hyper-filters with as few or many criteria as you want. You can go from a nearly empty filter (e.g. all fields clear except for **Folders** box that will find all folders) all the way to any arbitrarily complex filter with tens of rules combined in boolean contexts. Only items that match *all* aspects of the filter are considered hits.

 Name, contained text and additional property rules defined are normally combined in a boolean **AND** fashion (they must all be satisfied for a hit). Name and contained text can also be of boolean **NOT** type if you use a leading minus (-\*.PDF will match anything *but* PDFs). Additional rules can have all boolean contexts, even **OR** — see the little +/- icons in front of each rule. -


PS. try drag-dropping a file in the filter part of the find dialog and see what happens!

### Search workflow

When you press <CTRL+F> to start a new search, the find files dialog uses the same options and hyper-filter as the last time you searched for something. What changes is the **root** folder of the search, which is set to the folder you are currently browsing in the active pane.

If you want to do a new kind of search, first click **Clear** button to erase any traces of the last hyper-filter. Then manually add the new filter parameters or repeat a filter you [saved](#) earlier using the **Predefined** drop-down list. When you are all set, click **ok** button to start searching.

Searches can take a lot of time to complete. As soon as you initiate a search, a [scrap](#) window opens and shows matching files and folders as they are discovered. An active message pump allows you to start working with the first matches immediately, while the search is ongoing. You can *abort* a search by pressing the <Esc> key — or close the scrap window.

 A running search command slows down both the scrap and the main window, especially when using slow hyper-filters. Sometimes it will be better to **Window > Clone** a new main window before you start a search. This clone will run in its own thread and will be relatively unaffected and more responsive to user input. It is not advisable to start *another* search while you have one going, that's going to slow things down even more!


(ESC to abort); searching E:\music ...

You can check the progress of a search either by noticing the folder being scanned on the statusbar, or use **Tools > Search status** menu for a more detailed report, including all the folders that have already been examined, information about errors or **exclusions**, etc. This information is available even after the conclusion of the search. The search log is itself *searchable* using the context menu.

The scrap container with search results is a fully operational file management pane, so you can preview, open or manage all the files and folders found immediately — but wait till the search is finished if you plan to do any operation *en masse*. If you want to see a search result in a regular xplorer<sup>2</sup> window, right click on it and pick **Open file location** from the context menu.

If you searched for contained text, the draft [preview](#) tab will show you the keywords found **highlighted**. If you click in the preview pane and press <F3> key you will see the next keyword match, if any exists. If you used any fancy file properties in your hyperfilter, right click on the scrap pane list header row and use **Details from filter menu** command. This will make sure all properties you searched for appear, helping you understand the search results.



#### Further reading

- Can't find your documents? see the search [troubleshooter](#)
- Preview highlighting of search results [demo video](#) 


## Search in virtual folders

If you tick **Archived content** search option, xplorer<sup>2</sup> will also enter and search compressed archives like ZIP, 7Z, RAR and the like (if you have installed the [compressed folder](#) extension or a WCX plugin). If you are browsing within a zipfolder to begin with, then you don't even need to tick **Archived content** option, it is implied.


Many search parameters are valid in archive search, so you can search by name, size, dates and so on. Most of the [stock](#) properties are valid too. These special folder types support their own custom properties, and they can be used as search criteria as well! See for example a zipfolder which has columns for CRC-32 and compression ratio (pic to the right).

« C:\Users\god\Desktop\old desktop\newkiti - Copy.zip »					
Name	Size	Compressed...	Method	CRC-32	Ratio
 installer.jpg	58,628	47.0 KB	Deflated	4708FA86	18%
 readme.txt	279	193 bytes	Deflated	877EA0E8	31%

You can also search for contained text in compressed files. xplorer<sup>2</sup> temporarily extracts (decompresses) the searched files and searches them like "normal" documents. Obviously this will be a slow operation.

 Searching for contained text in archives will only work for files up to a size limit (see "*Byte size limits*" in program [options](#)), and in particular the limit set for "*Item in ZIP/FTP*". If you want to search for content in large archives, it may be quicker to first extract all files yourself, and then search in the regular folder created.

You can also search inside connected **phones** (which appear under *ThisPC* umbrella folder) using basic properties and even contained text. Searching in **cloud** folders is also possible, only xplorer<sup>2</sup> won't check for contained text in cloud-only (offline) items unless the [advanced option](#) "*Allow downloading on-demand cloud files...*" is ticked.

 When xplorer<sup>2</sup> extracts files from zipfolders, phones and such for searching or previewing them, several temporary files and folders are created in system %TEMP% folder. These have names beginning with x2tmp\_xxx. All these temporary files are deleted when you quit xplorer<sup>2</sup>

## Repeat saved searches

If you repeat a certain search often it makes sense to keep it for later. Once you search using <Ctrl+F> use **Tools > Save search** menu command in the scrap window to save the last search scope. This includes all the properties of the search dialog ([figure 43](#)): the starting folder, the search context (e.g. search subfolders and other flags) and the search filter.

Give a good descriptive name to each saved search so you can recognize it later. To repeat a search saved earlier, open a scrap window and use **Tools > Recent search** menu. Note xplorer<sup>2</sup> does not *cache* old search results, the search is repeated from scratch. If you want to save the *results* only, create a [CIDA file](#) using **Actions > Write contents** menu.

xplorer<sup>2</sup> installer registers the .X2FND extension for saved search files. So another way to start a saved search is to double click on a saved search file. xplorer<sup>2</sup> by default puts them in AppData folder but you can move them anywhere you like.

## Fast indexed search

Searching for stuff is ubiquitous in our modern computer life. xplorer<sup>2</sup> professional can find stuff on your desktop but it takes its time doing so. You need to search in a small targeted folder tree where you expect to find your documents. If you search your entire PC regularly, **professional** edition will be painfully slow.

[Desktop search](#) is a windows system component that accelerates file searches creating a **search index**, a database of file properties and contained text. When you create a new document, its properties and text contents are added to the database, making subsequent searches very fast indeed. Instead of searching folders one by one, hits are retrieved looking up this index.


xplorer<sup>2</sup> **ultimate** edition can tap in the **fast searches** available in windows explorer, querying the indexing server directly. You will notice a substantial speed increase especially if you search for text in the file contents.

#### DeskRule search tool


A specialized search tool by Zabkat combines fast and reliable file searches, in a fashion that is closer to internet keyword searches you are used to. It can also find photos by map (GPS) and files stored in disconnected storage. Help yourself to the [free trial](#)

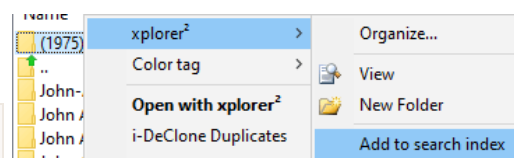
You don't need to do anything special to take advantage of this integration; when you press <Ctrl+F> to start a search, in many situations xplorer<sup>2</sup> will automatically retrieve results quickly from the system index in phase 1 of the search. It will then pause and allow you to review the results. If you think the file you wanted wasn't found just click on *robust search* button (you will find it at the very end of the results) to continue with xplorer<sup>2</sup> robust (albeit slower) search, folder by folder.

Not found what you were looking for?

 Click for robust search

To take advantage of fast searches you must make sure your work documents are indexed. This is easier if you store your stuff under the system Documents folder, which is automatically indexed. If you keep your documents in other locations, right click on the top level folder and use **Add to search index** command from **xplorer<sup>2</sup>** submenu. This need be done just once, then any new document you create in that folder (or its subfolders) will be indexed automatically.

 You can add any local folder to the search index. You can even add removable USB drives — but these need to be connected if you want to fast-search in them. Adding **network** folders in the index on the other hand is next to impossible. Network searches are condemned to be slow.





There are a few **limitations** to keep in mind. Searching for text in files will only find whole words and not partial keywords — at best you will find stems at the beginning of words. So if you search for *long* you will find files containing the words *Long* and *Longer*, but not those containing *prolong*. You can't search for special characters like brackets,

math symbols or quotation marks (they are ignored). Search index is good for word searches only.

xplorer<sup>2</sup> will **not** use the indexing server in the following occasions:

- When searching in folders outside the scope of the system index (e.g. removable or *unshared* network folders)
- When a local folder isn't indexed (add folders in the system index using the **Indexing Options** applet in control panel)
- When the search filter includes complex properties like regular expressions or fuzzy filenames
- When searching for file attributes (properties) not covered by the system index, e.g. the **Signed By** xplorer<sup>2</sup> stock column.

 When you search for mixed properties, only some of which are indexed, xplorer<sup>2</sup> will do a **hybrid** search. It will query the index for those properties that are supported, then do a slow search for the remaining non-indexed properties — but only for files returned by the index. This will lead to acceptable search speeds.

 Ultimate edition has yet another way to find files fast by **Name**. If you search only by name, and if you are looking for files in a local hard disk, xplorer<sup>2</sup> will use the NTFS Master File Table (MFT) to find the results very quickly. To use this mode you must be running with full administrator privileges.

Fast and simple search

A simpler way to access the system index is **Tools > Fast search** menu command. Instead of defining detailed search rules, you just type a keyword. xplorer<sup>2</sup> then starts a search under the *current folder* for any file or folder that contains this keyword in its name, text contents or any other searchable property (e.g. tags or author).

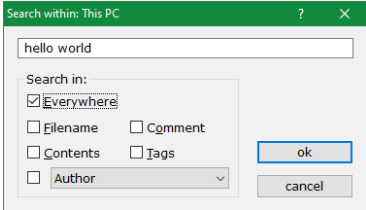




Figure 44. Fast search for files dialog

Tick **Everywhere** box if you want to find keywords in any searchable property. That is the simplest search mode. If you want to target something in particular like **Filename** or text **Content** use any of the other checkboxes. You can tick as many as you like. You can use custom indexed properties as well using the drop-down list (see **Author** in the above example).

 If you search for more than 1 keywords, **all** of them must be present somewhere, but in any order — this is how google search works. In contrast, if you did the same in the main [search dialog](#) without using commas, the exact phrase would be located — not partial words. In fast search format use quotation marks to search for "exact phrases".

Once you type your search keywords and tick the property options, click **ok** button to start the search. Like regular search a scrap window will open for the results. However here there is no backup robust search phase, files will only be found if they are indexed. If you see an error message on the status bar, make sure the folder you are searching within is in the search index.

 No items match the specified filter

## Automation of DOS (and windows) commands


In spite of the drag-drop and click-happy era of file management we're living in, for some tasks you can't beat the efficiency offered by traditional command line execution of programs. In the old days, instead of clicking on icons to start a program, one had to type its path/name in a command processor window (DOS console). This may seem quaint and antiquated but it is very powerful since you can specify extra *arguments* that modify the behaviour of the program in question, specify which files to operate upon etc.

xplorer<sup>2</sup> brings the command line into the third millennium, easing its use while maintaining all its power and flexibility. If you are familiar with the DOS console, you can regard the active pane cum addressbar combination as a graphical console that shows the folder contents of the "current directory" at all times. As you browse folders the "current directory" is updated automatically, and the commands you type can access the folder contents directly.

You can start composing text commands using **Tools > DOS command** menu, or just get into the addressbar with <Shift+Tab> and type the special prompt \$ character that instructs xplorer2 to interpret the input as a DOS command — otherwise it could be mistaken for a folder path. For instance typing `<dir` and pressing <Return> will execute the DOS **dir** command, producing the listing of the current folder's contents - arguably not the most useful command within xplorer2's context!

The commands you type correspond to names of executable files that exist on disk - minus the .exe or .com extension - and in the end of the day launch the respective application. So you must make sure you type the names correctly and that the files themselves can be *located*. If you have a program called windiff.exe located in c:\tools and you want to call it while browsing c:\work, then you must type its *complete path* in the addressbar, e.g. \$ c:\tools\windiff. Alternatively you can add c:\tools to the **PATH** environmental variable and reference its contained programs just by name

DOS commands create usually text output that goes into a DOS console window. xplorer<sup>2</sup> uses its own [user friendly](#) console window, but you can also use the traditional MSDOS console.

 TIP: if you want to open a MSDOS console window at the current folder location, type a solitary \$ dollar (without any command) in the addressbar and press <ENTER>. This window is then independent of xplorer<sup>2</sup> — just close it when you are done with it.

Except for DOS commands (\$-prompted) you can also execute generic windows programs using the special character **>**, e.g. typing **> winword** will launch MS Word. In this sense you have all the functionality of the standard **Start > Run** menu straight from the addressbar. This execution mode is mostly for programs that have their own user interface and don't need a console window for their output.

Except for launching both DOS and windows commands, the addressbar assists you during the typing process in a number of ways:

- It supports path [autocompletion](#) using the <F1> key. Start typing partial paths and press F1 a few times to complete folder and file names matching the stem
- <Ctrl+Enter> inserts the `[focused]` filename from the active view straight in the addressbar. Using <Ctrl+arrow keys> you can move the focus around while remaining in the addressbar, so you can insert more than one filenames as part of a command.
- Special [tokens](#) can be used in place of explicit command arguments. So instead of typing `> windiff file1 file2` that compares only these two files you can type `> windiff $n $i` which is reusable for many situations. All commands are available for future use through **Tools > Run history** menu.

Commands launched from the addressbar normally apply only to the single *focused* item in the active folder view. If you want to apply the command to **all selected** items use a double prompt character, e.g. >> or \$\$\$. Then instead of running the command once, it will be repeated for each selected item. This construct is a simple road to [scripts](#) without a dialog window.

## \$-Tokens

Instead of typing specific filenames as parts of a command, you can use special characters that get automatically substituted with the name of the item(s) that happen to be selected when a command is executed. For example the commands `$type file1` and `$type $N` are equivalent if `file1` is the focused item in the active folder pane. Using special tokens like `$N` saves keystrokes and allows for *reusable* commands. So whereas `$type file1` can only be used to type (show the contents of) `file1`, the version with `$N` can be used to type *any* file, as long as it is selected in the active view.

There is a variety of \$-tokens, each representing a different part of the active item or even the all selected items taken as a whole. Let's use a hypothetical situation to clarify the meaning of each \$-token. Suppose we have a dual-pane arrangement where the *right* view is active and the contents are:

- **Left pane:** (inactive) shows the contents of **c:\music** and contains 10 items, out of which only one is selected & active, called **titles.txt**
- **Right pane:** (active) shows the contents of **c:\work\c++** and contains 2 items, *both* selected; the first one is called **file.txt** and the second one **active.cpp**. The focused item — the one with the dotted rectangle around it — is **active.cpp**

The following table lists all the available tokens. The third column shows what each token would have been substituted for, given the above scenario. Unless stated otherwise all tokens act on the single focused (or active) item in each pane.

**Table 3.** Reusable command tokens


Token	Description	Example
\$N	Local filename	active.cpp
\$B	Base name (before .)	active
\$E	Filename extension (past .)	cpp
\$P	Parent folder path	c:\work\c++
\$D	File modification date	13/3/2004 10:12
\$C	Parent folder plain name	c++
\$F	Full path name	c:\work\c++\active.cpp
\$U	UNC path name (network accessible)	\\comp\share\active.cpp
\$S	All selected filenames	active.cpp file.txt
\$A	All selected filenames (fullpaths)	c:\work\c++\active.cpp c:\work\c++\file.txt
\$>	Like \$A but selected fullpaths are saved to the list file %TEMP%\x2tmpList.txt and this file is used as argument	%TEMP%\x2tmpList.txt
\$L	Left (top) pane path	c:\music
\$R	Right (bottom) pane path	c:\work\c++
\$I	Inactive pane path	c:\music
\$Q	Filename from inactive pane	titles.txt
\$G	Path and filename from inactive pane	c:\music\titles.txt
\$Z	Temporary filename extracted from zipfolder, FTP, etc Within normal filesystem folders it behaves like \$F	%TEMP%\x2TMP_001_active.cpp
\$nn	Automatically incremented counter starting from number <b>nn</b> Possible counter formats are \$1 for the sequence 1,2,... \$001 for 001,002,..., \$5 for 5,6,... and so on	01
\$[s:l]	Extract part of the filename starting at <b>s</b> and for <b>l</b> letters \$[4:3] will extract 3 letters starting from character 4 Use a negative <b>start</b> to count from the end of the name	ive (extracted from <i>active.cpp</i> )
\$?	Asks for an argument	(whatever you type)
\$	Escape for a single \$	\$

Many of these tokens are intended for renaming large numbers of files, as they are used in **File > Mass rename** command. But you can use them for any other purpose too; just type as many tokens as necessary in a single command and they will all be substituted according to the above rules.


Now we are in a position to explain the celebrated windiff command used to compare the focused item in the active pane to its namesake in the inactive pane:

```
>WINDIFF "$N" "$I"
```

Consulting table 3 above, \$N corresponds to the focused item, its full name without path, and \$I is the path of the inactive pane (without explicit filename). Note we enclosed tokens in quotation marks to guard against spaces in paths and filenames. This generic command template can be used to compare whichever file is active — and regardless which pane is active (left or right).

 \$Z token can be used to extract files from compressed ZIP archives. It can be used to compare files in ZIPs with those in regular folders. Make sure the zip folder is active, and use this command:  
windiff \$z \$g  
\$z will stand for the temp file extracted and \$g is the focused item in the inactive (regular folder) pane. Note you must select the correct file in the *inactive* pane as well, perhaps using **Mirror scrolling** command <CTRL+M>. \$I wouldn't be useful in this situation as the extracted filename is somewhat mangled with x2TMP\_xxx stem.


You may have noticed that all tokens in table 3 were listed in uppercase. If you type a token in lowercase then it is interpreted in a slightly different fashion, using the short 8.3 version of whatever name the normal uppercase token would have produced. To clarify, had \$N corresponded to **undocumented.h** the lowercase \$n would have resulted in **UNDOCU~1.H**, which is the equivalent old DOS filename.

 8.3 filenames look obscure but they don't contain any spaces so they can be used in commands without worrying for quotes. In some partitions 8.3 names are disabled (they usually exist for C: but not for extra partitions). In that case xplorer<sup>2</sup> will add quotation marks to protect against spaces in filenames, when lowercase tokens are used.  
Tokens that correspond to multiple items like \$A automatically add quotes as necessary (even in uppercase format)

Another kind of token can be used to extract the text of a particular file [column](#). The generic format is \${column\_name}, e.g. \${size} will fill in the file size as it appears in detailed view mode. You can use any of the available columns, which comes handy for [mass renaming](#) as e.g. you can use MP3 tags as part of the filename.

For date property tokens you can supply your own format specifier, overriding the way dates appear in columns. This is most useful when [friendly dates](#) are in force. You can supply a custom date format as \${Modified:yyMMdd HH:mm} where date and time [format pictures](#) come after the property name, separated by a colon :

The colon modifier has an application for counter tokens too, signifying a **step size**. So whereas plain \$01 generates the sequence 01,02... the modified version \$01:3 would generate the sequence 01,04,07... (numbers increasing by 3).

 The addressbar in [scrap](#) windows also supports command execution. The only difference is the concept of the "current directory". So whereas in normal browser windows the current directory is synchronized with the active folder, scrap windows contain files from many folders and as a result there is no current directory in the DOS sense. Therefore you should only use tokens that carry full path information, e.g. \$F instead of \$N, otherwise you may have problems addressing the intended files. The tokens \$L, \$R and \$I don't have any meaning for scraps altogether.

## Command output redirection

Most DOS commands don't have a GUI of their own and rely on the console window for their output. This is the traditional black & white system DOS box, which is rather awkward to use. xplorer<sup>2</sup> comes with a substitute console that is nearly equivalent in functionality and much easier to work with, e.g. it can mark and copy text like a normal editor window, **search** for text, etc.

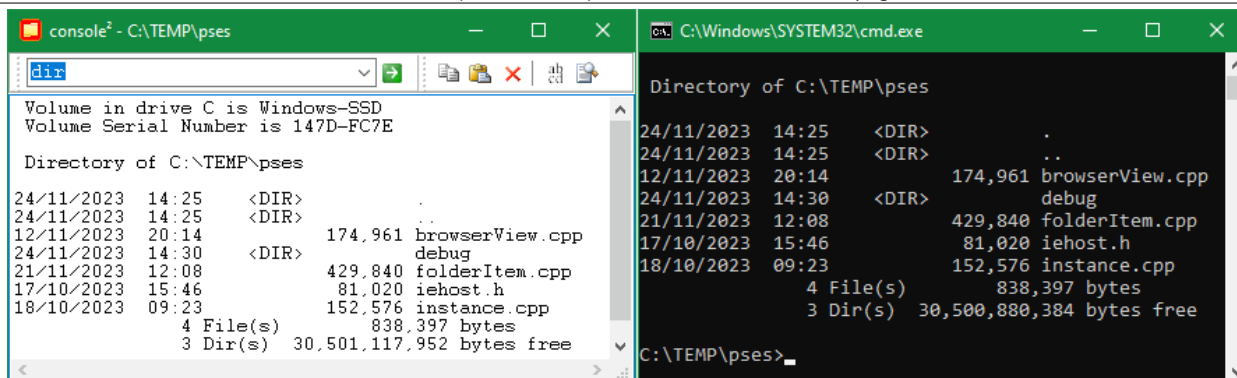



Figure 45. User-friendly vs traditional DOS console

This modern version of the DOS console automatically follows the browser window, changing its current directory as you browse folders with xplorer<sup>2</sup>. The working path is shown on its title bar. It receives the output of all \$-prompted commands issued from the addressbar and from command [scripts](#), too.

The top row **toolbar** lets you easily copy/paste and even search for text in the console output. You can also select the font to use.

You can also type commands directly in the console window - that's the purpose of the input field ("addressbar") shown in figure 45, as well as deliver input to running programs (e.g. Y/N responses). This input area supports path autocompletion with <F1> key and maintains a history of past commands too.

 The xplorer<sup>2</sup> console doesn't have a "prompt" line, but you can type extra commands in its addressbar — they will operate on the current directory (folder) you browse in the main xplorer<sup>2</sup> window. You can also type anywhere in the console window and press <ENTER> key to execute commands. This is a bit weird, but it works!

If despite all the extra functionality you prefer the traditional DOS console, you can disable it unchecking "Redirect DOS command output..." program [option](#). Note that in such a case you'll get a separate console for each DOS command you execute.

## Automatic script generation

Commands launched from the addressbar operate only on the single *focused* item within the folder being browsed (unless a double prompt as >> is typed as discussed above). In some occasions you may need to apply the same command to a number of files as for instance for batch conversions of MP3 files. For this task it is more convenient to use **Tools > Command script** menu to treat *all* selected files in one stroke.

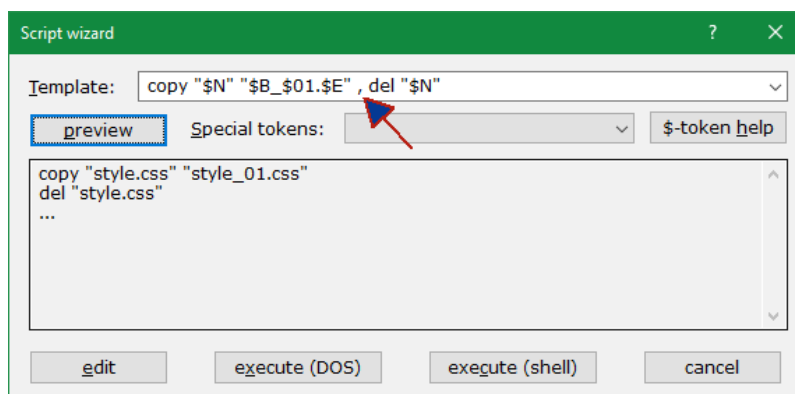


Figure 46. Dialog for batch file generation

In this dialog you specify a command **Template** that will be applied to each and every *selected* file name. The syntax is identical to the addressbar commands, using the familiar [\\$-tokens](#) instead of actual file names. Command templates may contain *multiple* comma-separated commands as the above snapshot illustrates. If you want to add a comma verbatim in a command escape it with two `,,` commas.


If you don't remember the token names to use, you will find most of them under **Special tokens** drop down list. For extended information click **\$-token help** button. As you compose your command template use the **preview** button to double check the command looks like what you intend it to! It will apply the template on the *first* selected filename and show the command(s) in the preview window. The example above inserts a counter at the end of the filename by a laborious route (copy then delete instead of a simple `REN`) — it was selected for demonstration rather than for its usefulness :)

The script generator replicates the command template for each selected file in succession. Plain text parts are reproduced verbatim and \$-tokens are replaced on a per-file basis. If you have 100 files selected then 100 sets of commands will be generated, effortlessly. Let's have an example using the template `ren \"$N\" \"$B_$01$$. $E\"` on a selection that comprises 2 files, `x2help.htm`, `x2help.css`.

This template has four special \$-tokens, **N** for the whole name, **B** for the base, **E** for the extension and an automatic counter starting from 1 with a leading zero (\$01). The double dollar is an escape for a single \$. For each file the generator copies the constant parts of the template including spaces and quotation marks, and substitutes the variable parts using the file in question. For our two files this translates to (notice the counter in the second line):

```
ren "x2help.htm" "x2help_01$.htm"
ren "x2help.css" "x2help_02$$.css"
```


The commands are executed on each item in the *order* they appear in the active pane. Most of the time this isn't important unless you use automatic counter tokens; in this case make sure your selected items are sorted in the appropriate sequence before executing the script.

 The use of quotation marks around the filenames covers for situations where names contain spaces. An alternative is to use lowercase token versions like **\$n** that are guaranteed to be free of spaces, albeit rather mangled.

Once the script is generated, there are a number of alternative modes of execution, each activated by a dialog button:

- **Execute (DOS)**. This is the natural mode, running a batch file in DOS. Behind the scenes a file called **x2tmpScript.bat** is created in the %TEMP% directory.
- **Execute (shell)**. This is the script equivalent of [>-prompted](#) windows program execution. In this case templates cannot contain commands like `ren` that require the DOS interpreter.
- **Edit**. Instead of immediate execution you can edit the script file, perhaps adding a few final touches. If you are in dual-pane mode the inactive pane will focus onto **x2tmpScript.bat** so that you can easily execute the final script by double-clicking, after you save your changes.

You should note that the wizard is merely manipulating strings and is oblivious about their meaning or command syntax. If you make any mistakes you will notice during execution! Always preview the template before executing the commands.

 Batch scripts using DOS commands are very last century. Nowadays administrators are more likely to use [powershell](#) to work with files on the command line. xplorer<sup>2</sup>



scripts can be used to execute powershell commands on selected items, as long as you enclose everything in powershell -command "...". Use full path tokens to address files as xplorer<sup>2</sup> current directory won't be transmitted to powershell. Here is an example:

```
powershell -command "& {(get-item '$F').LastWriteTime = get-date '${Picture date}'}"
This uses $-tokens to change the modified time of selected JPGs to the date the photo was taken!
```

#### Further reading

Advanced file management using windows scripting host ([WSH](#))

## Favorite user commands

Instead of typing commands in the addressbar, you can store your favorite commands in a list for quick and frequent access. The easiest way to add a command is to first find the executable program (browse its container folder and select the EXE file), then use **Customize > User commands > Add current** menu command. This will pick the executable's full path and you just add whatever command line arguments you need and store it in the list.

For more customization use **Customize > User commands > Organize** menu, which will open a dialog like we have seen for organizing [bookmarks](#). Edit the user command you just added to change the icon, add a keyboard shortcut for quick access and so on:

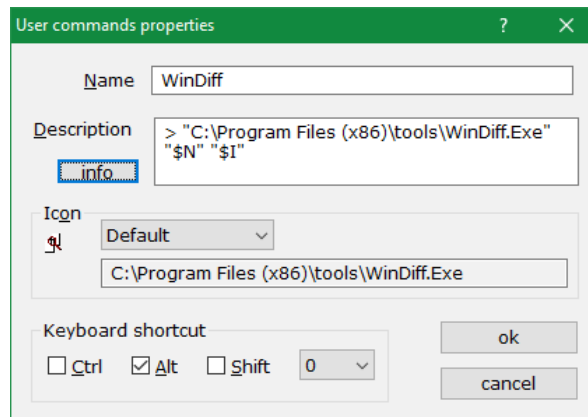


Figure 47. Setting individual user command properties

The **Name** is just a mnemonic for your own use, as it appears on the user command list. The important part is the **Description** which is the command line, including the program and the arguments. Notice you must start the command with the correct prompt character (\$ for DOS and > for windows GUI tools). Click on **Info** button to see a reminder of available \$-tokens.

The "description" field is rather narrow for most commands with rich arguments. But never mind, long commands will wrap in multiple lines. As long as you don't add newlines, you should be fine. You cannot use commas to define multi-line commands either.

For really reusable command templates, you can use environmental variables like %WINDIR% in the command paths, these are recognized and substituted for the correct full path before execution. The special environmental variable %X2DIR% stands for the full path of xplorer<sup>2</sup> installation folder, and can be used for [portable](#) commands, assuming you save your extra tools where you keep your ultimate xplorer<sup>2</sup>.

Once you define a command, you can access it from **Customize > User commands** list. For easier access assign a keyboard shortcut to it, or add it on a toolbar through [customization](#). User commands in toolbars are *drop targets*, meaning you can drag drop a file on a command button and it will open in the respective user command.

The *order* of command items determines their position in the respective submenu. Items in the top of the pile are easier to access so you should keep the most frequently used ones near the beginning (use **organize** command to reorder them). Each submenu shows up to 20 user commands; however you can define up to 100 items for each category and access them using the **More...** menu command.

Here are some examples of really useful external commands. The first uses the \$A token to join multiple selected PDFs in a single file called OUT.PDF. The second can convert any video file to the MP4 format. Obviously you need to download the tools first and change the paths to match your installation.

Table 4. Useful external commands

Name	Description
join pdfs	\$ "C:\Program Files (x86)\tools\pdftk bin\pdftk.exe" \$A cat output out.pdf
convert to MP4	> C:\PROGRA~2\tools\FFMPEG.EXE -i "\$N" "\$B.mp4"

## Optional command line arguments ▶ PLAY

When executed from the command line xplorer<sup>2</sup> accepts a number of optional arguments that control some aspects of the program. The complete command line is:

```
xplorer2.exe /F:n /P /R:rootFolder /S:registryKey /L:searchFromFolder /I:settings.REG /M /1 /2 leftFolder rightFolder
```

All arguments are optional and are described in the following table. The last two arguments — without the / switch character — specify starting left & right pane folders for the main window. These can also be the names of folder groups saved in **Customize** menu if you want to reinstate a particular set of tabs upon startup.


If you start with a scrap container (/F:1 switch) *leftFolder* can be a [cida](#) file or a name of a folder to flatten.

Table 5. xplorer<sup>2</sup> command line arguments

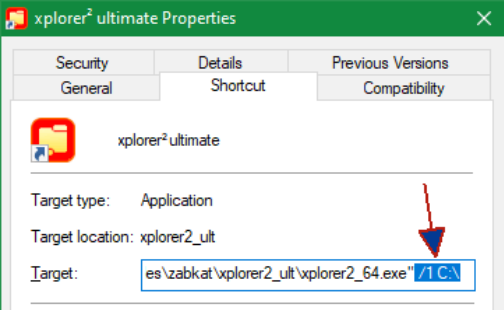
Switch	Function
/F	Starting window type. By default a normal browser window is opened, equivalent to /F:0. If you want to start with a scrap window use /F:1
/P	By default xplorer <sup>2</sup> runs in a system resource-friendly single process mode, even when you launch separate instances. Using /P you force new instances to open as separate processes.
/R	Roots the program on a folder other than the default desktop. For instance /R:c:\winnt will force both the tree and view panes on c:\winnt and won't allow users to reach other parts of the namespace unless underneath the root node; e.g. d:\ will be off-limits whereas c:\winnt\system will be accessible
/S	Registry key extension for storing program options. The default key is HKCU\Software\ZabaraKatranemia Plc\xplorer2 but you can have multiple parallel keys with different options. Keys can be changed dynamically with <b>Window &gt; Save layout</b> menu. For instance /S:preview will load options from the key ...\\ZabaraKatranemia Plc\xplorer2_UC.preview
/L	Causes a scrap window to issue a "Find Files" command when it first comes up. The search is rooted at "searchFromFolder"; alternatively it could be a saved search .X2FND file. Can only be used in conjunction with /F:1
/M	Does not allow window to open in minimized mode on startup
/N	Do not restore slow folders from last session (e.g. network) in case they are unavailable
/1	Forces single pane mode
/2	Forces dual pane mode
/T	Doesn't restore folder tabs on startup (by default you get all tabs reopened - as you left them at last use)
/D	Force xplorer <sup>2</sup> to nag you for a registration key
/I	(portable version) Initialize registry with a custom settings file. The settings file should be in the same folder as xplorer <sup>2</sup> and be created with <b>Actions &gt; Extract settings</b> on your "home" PC. For <b>ultimate</b> version this can also be a INI file bypassing the registry
/E	Run in windows explorer replacement mode
/Z	Silently start and stay in the background without a window
/B	Execute one of the saved macros (see <b>Customize &gt; Macros</b> menu) when xplorer <sup>2</sup> starts. For instance "/B:sample macro" will launch the saved "sample macro" (assuming the name exists). This switch is useful for automating tasks using the task scheduler

You can take advantage of these options by creating a desktop *shortcut* for xplorer<sup>2</sup> (or use the one added by the installer). Right click on the shortcut icon and pick **Properties** from the context menu. Switch to **Shortcut** property page, and add the required command line arguments in the **Target** field, past the executable name. You must leave a *space* between the executable and each option you add. The snapshot to the right illustrates a shortcut that launches the program using the options /1 C:\, which means start with a single pane and browse folder c:\

Likewise you can create multiple desktop shortcuts, each for a different starting folder or starting *layout* (passing the layout name with /S: switch). Rename these desktop shortcuts to recognize what each one does.

 if the folder path you want to pass as an argument contains **spaces**, enclose it in "quotes". Likewise, if a layout name contains spaces, enclose the entire switch in quotes, as such:

```
xplorer2_64.exe "c:\some path\with spaces" "/s:layout name"
```







## Advanced programming and scripting

One step up the ladder from DOS scripts, come user interface macros and file property algebra. These advanced topics are closer to computer programming than casual file management, but they don't require a degree in computer science to take advantage of. If you are used to MS Excel cell expressions, you should get the hang of it quickly. Note that xplorer<sup>2</sup> doesn't offer a fully scriptable programming model with file and user interface objects, but a simpler macro language for user interface elements, that should be within the reach of xplorer<sup>2</sup> medium-to-advanced users. As for the programmers amongst you, I refer you to the [windows scripting host](#) for the full monty <g>

### User interface automation with macros ▶ PLAY

A simple in-app "programming language" allows you to automate many program tasks. It is not a full blown scripting language (as e.g. the one in MS Excel) but it is surprisingly versatile and easy to use. You write **scripts**, that is sequences of simple text commands that manipulate the program window as if a robot was in control. You can do these sorts of tasks:

- Execute a command that launches a dialog
- Find windows (dialog control elements) like buttons, text fields etc
- Set text in edit controls
- Check radio buttons
- Select from lists
- Click buttons
- Wait for commands to finish

All in all you can do whatever any user could do using the keyboard and the mouse to operate the program, only do it in a tightly controlled fashion. You can string commands together that operate on a series of dialogs to achieve complex results. Many times you can get a head start using the [macro recorder](#), where you just use the program for what you intend to capture with the macro, and get a skeleton implementation for free. However first let's do it the hard way, writing everything from scratch. Here is an example script:

```
# copy selected files with filter
COMMAND "copy to"
WAITWND
FINDWND filter
FINDCLASS combobox
SETTEXT *.pdf
OK
```

As you can tell from the command names, this macro is equivalent to clicking on the **Copy To** toolbar button, waiting for the dialog to appear, then finding the [Filter box](#) and setting its value to "\*.pdf"; and finally "clicking" the OK button to start copying the files with the selected options (only PDF files will be copied).

Use **Customize > Macros > Edit macro** menu and copy/paste the above script in the editor area as such:

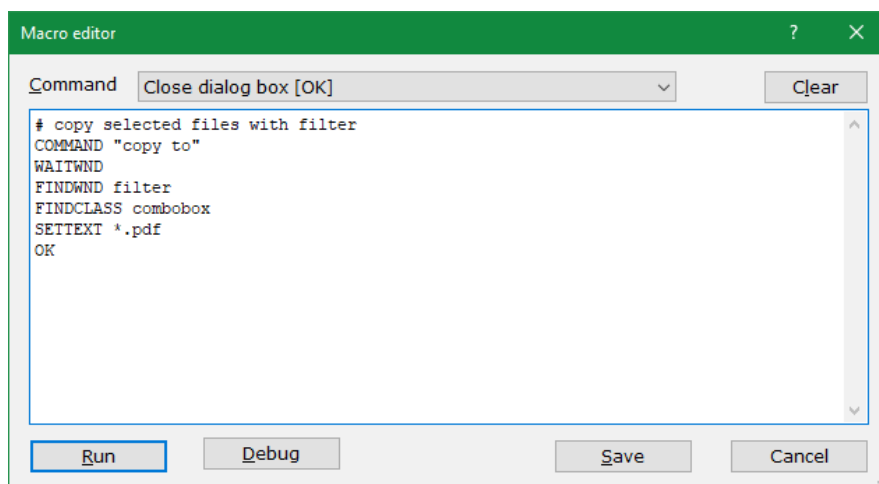


Figure 48. Macro editor: compose new macro

Going quickly through the macro editor dialog, we see a **Command** drop-down that lists the most popular commands, which you can insert with guidance on the expected arguments. **Clear** button erases any previous macro so you can start fresh. When you are ready, click **Run** to execute the macro, or **Debug** button to [troubleshoot](#) problems. **Save** will add the current macro in **Customize** menu so you can use it again in the future.

Click **Run** to run the above macro. If you had any PDF files selected, it will copy them to the inactive pane, otherwise it will do nothing. When a macro is running you cannot work with the main program window. You will see dialog boxes opening and closing in auto-pilot mode, following the macro commands. A macro control window appears on top of other windows that shows the progress of the execution. To abort a macro mid-way through its execution click on the **Cancel** button.



While a macro is running you **should not** switch to another window or application, otherwise the macro will probably fail. Nor should you interfere with random clicks or keypresses. Just hang in there till the macro is finished!

When you repeat a macro saved in **Customize > Macros** list, it works in **silent** mode; you won't see any dialogs popping up, just the result of the macro instructions. When the macro finishes, you will see a message on the statusbar — unless some error steals the show!

**Macro command finished**

To abort a silent macro that seems stuck press <CTRL+BREAK> to stop it — if you can find the key!

### Macro command syntax


Each line of a macro contains just one command and its arguments, if any. For example:

```
COMMAND "copy to" # start the copy to dialog
```

**COMMAND** is the reserved command name that tells the macro what to do, "copy to" is the argument (which program command to execute), and the rest is the comment. Comments are ignored by the script but help you to make sense what each command is doing. Inline comments begin with the hash # symbol.

The macro language instructions and searched command and window labels are all case **insensitive**. You can use any case you prefer, e.g. command "COPY TO" is just as valid as the above example.

Arguments are necessary for many commands. They are usually **constant** text strings as the name of the toolbar button you are trying to click. Use "quotes" whenever you have spaces in arguments (many words). Arguments can also be numbers e.g. 32793 is the equivalent identifier for "Copy to" toolbar button (use **Help > Command finder** menu to find numeric equivalents to [commands](#)). Most of the time it will be easier to use the text label, but sometimes there will be no alternative than to use the number, e.g. for hidden commands. Arguments can also be variables, which will be discussed below.

 Commands names should be searched as they appear in xplorer<sup>2</sup> menus or toolbar. If you are using a translation, search commands in your native tongue; using english names will not work in this case. For language-independent macros use command ID numbers instead of their names.

Many commands return a value, e.g. `FINDWND` returns the handle to the window found. You don't see this result immediately but it is stored in the **last result** register (\$0). Subsequent macro commands may rely on the previous result to function properly, e.g. `CLICK` needs a window handle found earlier. The last result is piped from one command to its successor automatically.


9 elementary **variables** are supported to be used as arguments, called \$1 to \$9. Most of the time you will use \$0 if you want to visualize the result of the last command (`MESSAGE $0`).

## Finding windows and menu commands


The most important operations you will be using all the time is finding control windows (buttons, edit controls etc) to operate, and issuing message commands (using menu or toolbars) to the program — using `FINDWND` and `COMMAND` respectively.

If you are familiar with windows programming, you will know that each menu command and toolbar button appearing in a program's user interface correspond to a **message number**. Likewise, controls that appear in dialogs have an **identifier** number. For instance 32793 is the message number corresponding to "Copy to" toolbar button; and the OK button in most dialogs translates to number 1. Unless you are a big time geek <g> that can use Resource Hacker or Spy++ you cannot be expected to know these numbers!

Laymen needn't worry. Instead of fumbling with cryptic numbers, you can use the text that appears on a menu, toolbar or dialog button to tell the macro which command you are after. Take note which text appears on the menu command you want to issue, and use it as the argument to `COMMAND`. For toolbars use the **tooltip** if the button you are after has no text showing. It is best to use the **full** command text as it appears in the GUI, which will ensure that the intended command will be issued. Partial names can be used too, but keep in mind that the first matching name will be used, which may be some other command or window! All

 **TIP:** When troubleshooting macros, insert a `MESSAGE $0` instruction under a `FINDWND` or `COMMAND`. This will show you which window or command was found (\$0 is the last result variable). Once you are done debugging you can delete or comment out these `MESSAGE $0` instructions. Alternatively you can run the macro in **debug** mode where you single step the execution and see detailed information about each command, see which window is operated upon etc.

Finding buttons on **dialogs** is easy, just use the name of the button with `FINDWND`. Other controls have some descriptive text label before them, which may be used to locate the control. Sometimes a control has no useful text or label; in these cases locate the control **nearest** to the unlabelled one, then move forward or backward with `NEXTWND` or `PREVWND` instructions till you hit the desired dialog control. The directions "forward" and "backward" correspond to the tab order in the dialog — or in plain english, what happens when you press <TAB> key to move around in a dialog window.

 When searching for commands and windows, keep in mind that the search is within the **active window**. If no dialogs show, that means the main program window; if a dialog is showing, the search is within the dialog controls. Use the wait instructions (e.g. `WAITWND`) to make sure the window you want to manipulate has had time to show up.

## Dealing with syntax and execution errors

When you first start working with macros, inevitably you will make mistakes. For example you can misspell an instruction name or forget a necessary argument. In that case a **syntax error** will result and the macro editor will highlight the token that caused the trouble. Here are some pointers to keep in mind:

- One macro instruction per line. Don't use ; to terminate commands
- Some instructions expect arguments, some don't
- If your arguments contain spaces, enclose them in "quotation marks"
- Are there matching unique labels for all jump instructions?

Once the basic syntax is correct, you can try running the macro. Here you may encounter subtle problems that will cause execution errors. The macro will stop, highlight the command that faulted and show an error message. Here are the runtime **error codes** explained:

- **1: unknown?**: mystery error code probably caused by an error message during executing the command in the GUI
- **2: Wrong argument**: you passed an empty variable to a command that expects an argument
- **3: Search failed**: a command or window could not be found (or is disabled)
- **4: Wrong window class**: you tried to operate a window in an unsupported way, e.g. change the text of a button or click an edit control
- **6: Command disabled**: the command or window cannot be manipulated because it is disabled

Error code 2 (wrong argument) is raised also when an instruction cannot find the correct last result from the previous command. For example you cannot use `CLICK` without a preceding `FINDWND`. Most commands that operate on dialog controls require a window handle in the last result \$0.

When you get an unexplained error 3 (not found), it could be that the dialog didn't have time to appear. If you use `COMMAND` and expect to operate the dialog controls, use `WAITWND` to make sure that the dialog is visible before you start manipulating its controls. `FINDWND` instruction is searching the dialog that is currently active (excluding the macro dialog itself).

Finally you may realize that a running macro is doing crazy things that were not intended. In that case click on **Cancel** button to abort the macro execution.

## Debugging macros

The simplest problem you encounter is a **syntax error**, where you make a spelling error to a command name or forget to close a quotation mark. These errors are easily corrected in the macro editor (see the previous section). Somewhat harder are runtime errors that reveal something's wrong with the logic of your macro.

When you run a macro and it errors out, you are offered to run it again in debug mode. Or click **Debug** button in the [macro editor](#) to enter the debugger.

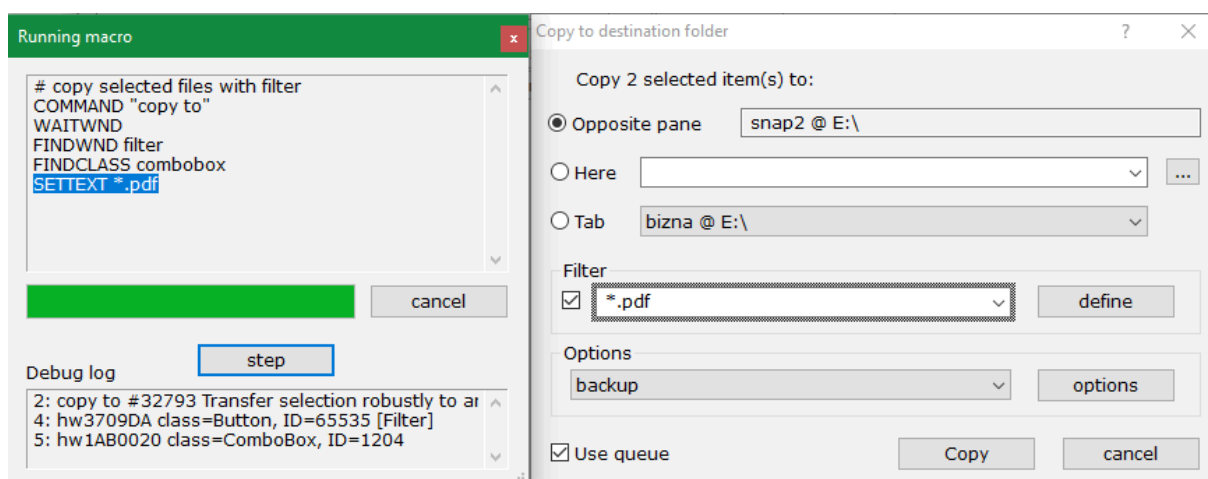


Figure 49. Running a macro step by step

In debug mode, the running macro (modeless) dialog is taller — you can resize it to taste. Instead of running the macro wholesale, it runs one line at a time. The current instruction is highlighted. Click on **step** button to proceed with the next instruction. Each command shows a little information in the **Debug log** area which you can examine to see if it does what you meant to do.

A cool feature is that it will **highlight** the last dialog control found — notice the thick border around the **Filter** combobox in the above picture.

It isn't quite in the same class as WINDBG, but with all the information shown, you should understand what the problem is, and go back to the macro editor to correct it.

## Conditional execution

By default all execution errors are terminal and stop the macro dead. However you may want to take responsibility and manage the execution despite previous errors. For example, if a control cannot be found you may want to try to search for another. If you want to ignore all errors, use the instruction: `ERRORSTOP 0`.

Once you turn off the automatic error handling, the last result `$0` variable will reflect the execution status of the last instruction. You can test for success using the `JUMPOK` instruction, which will jump to a label (i.e. move the point of execution) only if everything goes well — otherwise the instruction immediately below it will execute. Here's an example that looks for a window called *test* and stops execution manually if not found:

```
ERRORSTOP 0
FINDWND test
JUMPOK found
MESSAGE "window not found"
EXIT
LABEL found
CLICK
```

Using conditional jumps and `GOTO` instructions you can modify the order that script lines execute. Instead of linearly executing all lines in sequence, you can jump to earlier or later lines depending on your logic. Jump destination points (`LABEL`) can be any text you like to make sense of your script. If you have multiple jump points, make sure each label gets a unique name.

Simple **loops** are also possible using a control variable as a countdown counter, as such:

```
POP 5 # repeat 5 times
STORE $1
LABEL repeat
MESSAGE $1
DECR $1
JUMPOK repeat
# end of loop
```

The trick is that `DECR` instruction updates (decreases by 1) the counter variable `$1` *and* the last result, so when it reaches zero, you break the loop (`JUMPOK` fails). Also note the roundabout way to initialize the counter variable using `POP` then `STORE`.

## Macro command reference

The macro composition dialog has a listbox with the most common commands so you don't need to remember the names. Below you will find the full list of supported macro instructions and their expected arguments.

A typical instruction accepts a single **argument**, but there are some that apparently work on thin air without arguments. This is because they use the result of the previous command as their argument — this is called **pipe** argument. For example `CLICK` operates on the result of an earlier `FINDWND` command. If a command fails the result is zero (void).

Arguments can be either constants (strings or numbers) or variables (`$0` to `$9`). Variables represent strings, numbers or window handles, depending on what is stored in them. The last result is kept in variable `$0`. Unless otherwise stated, the following table uses **name** for either a string or number, furnished either as a constant or variable.

Harnessing macros will take some practice. Below you will find some sample macros you can play with. Try running them in debug mode (line by line) to see what they do with xplorer<sup>2</sup>. Observe the inline comments for extra explanations. (see [macro samples](#) below or [here](#) extern)



Table 6. Macro instruction reference

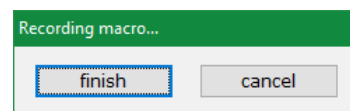
Instruction	Argument	Pipe arg.	Result	Description
FINDWND	name		window	Find a window by <i>name</i> on it or next to it (if labelled). Searching for part of the name is ok if you are sure that it will uniquely match the window. Also finds by window numeric <b>identifier</b> . The search is in the topmost active dialog or the main window. FINDWND 0 returns the currently active dialog.
FINDCLASS	name	window	window	Find a window by its exact <a href="#">class name</a> (e.g. edit, button etc). If the prior result is a window, it finds the next window of the same class, if any, possibly cycling from the top.
FINDFRAME	title		window	Find a top level window by its titlebar text and switch to it (make it active). The window must be selectable, e.g. not blocked by a modal dialog.
GETFOCUS			window	Returns the active window (where you last clicked).
SETFOCUS		window		Activates the window
NEXTWND		window	window	Returns the next window in the tab order
PREVWND		window	window	Returns the previous window in the tab order
CHILDWND		window	window	Returns the first child window (of whatever was found earlier)
PARENTWND		window	window	Returns the parent window
CLICK			window	Clicks on the window found earlier. May fail if the button is disabled
CLICKDEF	name	window		Executes the default accessibility command for child <i>name</i> (if using a number, start from 1). This is for non-standard buttons that don't have window handles (exposed through MSAA). For many standard controls it's equivalent to double-click on a list item.
SETTEXT	name	window		Sets the window text to <i>name</i> . Only meant for edit controls and combo boxes.
ADDTTEXT	extra	window		Adds the <i>extra</i> text to the existing edit control text, at the current insertion point. In contrast, SETTEXT replaces the entire window text, discarding the old contents
GETTEXT	\$var	window		Gets the text of the last result window and stores it in variable \$0 - \$9
GETCOUNT	\$var	window		Gets the number of items in listview control and stores it in variable \$0 - \$9
GETITEMTEXT	index	window	text	Extracts individual item text from windows like listview controls. Items should be specified by a number from 1 to GETCOUNT.
SELECT	name	window		Selects an item in a combo, list or listview control either by name or by number (1 → max items present). SELECT 0 will unselect everything
GETSEL	\$var	window		Gets the selected item number in selectable controls and stores it in variable \$0 - \$9
SETCHECK	1/0	window		Checks or clears a checkbox or radio button, if enabled. Special value 2 means flip previous state
OK				Terminates the active dialog. It is a combination of FINDWND "ok" followed by CLICK. It may fail if the dialog won't close due to validation errors.
CANCEL				Simulates pressing ESC key. If a dialog is active this will be equivalent to clicking on Cancel button
COMMAND	name		help text	Issues a program command (WM_COMMAND) by number or by name. It is equivalent to using a menu command or toolbar button. Searching by <i>name</i> is more intuitive; it finds the command from a menu or toolbar that corresponds to the (partial) name. For toolbar buttons without names use the tooltip (popup) name. If the active window is a dialog, it finds the named button and simulates a click. If the command is disabled (e.g. ghosted toolbar button), it fails. If successful, returns the explanation text of a the command issued (use MESSAGE \$0 to see it).
CMENU	name	window	number	Simulate a context menu (right click) on the window last found, and find the menu command that corresponds to <i>name</i> . Returns the command identifier of the menu item found. Can be used for the shell context menu or any other background menu
WAITWND				Wait till the active window changes, e.g. a new dialog window opens
WAITIDLE				Wait till the active window finishes processing the last command.
WAIT	period			Wait a fixed amount of time in milliseconds (1000 = 1 sec). The argument may also be a window handle in a variable, where it behaves like WAITIDLE for that particular window.
ERRORSTOP	1/0			Turns the "resume on error" policy on/off (default is to stop on any error).
LABEL	name			This instruction doesn't do anything but marks the destination of a jump/goto instruction
GOTO	label			Unconditional jump to label <i>name</i>
JUMPOK	label	any		Conditional jump to label <i>name</i> . Checks the result of last command and if it is failure/empty/0 then does nothing. So it only jumps if the last command was successful
JUMPERR	label	any		Conditional jump to label <i>name</i> . Checks the result of last command and only jumps if it was 0/failed/empty — the opposite of JUMPOK
EQUAL	name	any	0/1	Compares its argument with the last result. If the 2 values are compatible and equal, the result is 1, else 0
DECR	\$var		\$var-1	Decreases the variable (\$1 - \$9) by one and updates last result too. Used for loop control.
INCR	\$var		\$var+1	Adds 1 to a variable (\$1 - \$9) and updates last result too.
MATH	expr		number	Calculate the <i>expr</i> argument and store the integer result. Allows performing simple integer arithmetic e.g. (\$1+3)/2
POSITIVE	\$var		0/1	If variable is positive (>0) returns 1 else 0. Meant for integers mainly.
EXIT				Stops execution; any instructions below this are ignored
JSR	label			Unconditional jump to label <i>name</i> as a <b>subroutine</b> . There is no stack and no nesting is supported. Use regular \$0-\$9 variables for information exchange
RET	code		=code	Resume execution at line following the last JSR instruction. Used to return from a "subroutine" with a code inserted in \$0
MESSAGE	name			Pauses execution showing a message. Debugging aid mainly used to visualise the result of the last command as MESSAGE \$0
ADDSTR	name	old	old+ name	Adds name to the old string result. If you need a newline use "\n". Intended for composing longer messages. With an empty argument it resets (empties) the string result.
CHOPSTR	format	full	part	Splits the last result string into tokens and returns one of them. The format argument includes the split character and the substring number to return. For example \2 would separate a path on \ points and return the second part; \t-1 would split on tab characters and return the last part (negative numbers count from the end)
REM	any			Comment line that doesn't do anything but explain something. For inline comments use #
INPUT	prompt		text	Pops up an input dialog where you enter a string, which is then stored in the result. Prompt is used as the question to ask. Tip: use INPUT 0 to import the text from the clipboard.
STORE	\$var	any		Stores the last result in a variable \$1 — \$9 for future use
POP	any		=any	Copies the constant or stored variable into the last result \$0 register to be used in a subsequent command
SENDKEY	keys	window		Send keyboard input to a window. E.g. CTRL+LEFT will send a combination of control and left arrow keys. Most special keys are recognized by name (see <a href="#">VK_ xxx names</a> ), and for the rest you can use the equivalent virtual key code number, e.g F1= <b>112</b> (from hex 0x70)
KEYDOWN	key			Simulate a single key press down (and hold). Unlike SENDKEY a window target isn't required as pipe argument, the key goes to whatever window has the focus. Can be used to simulate <SHIFT> combined with button clicks in dialogs.
KEYUP	key			Simulate a single key release. Each KEYDOWN should have a matching KEYUP or the keyboard will go funny.
NOPOPUPS	1/0			Turn silent mode on/off, i.e. don't show any of the dialogs that would normally appear during macro execution. This mode is on by default if you repeat a macro.

Harnessing macros will take some practice. Below you will find some sample macros you can play with. Try running them in debug mode (line by line) to see what they do with xplorer<sup>2</sup>. Observe the inline comments for extra explanations.

• [Click here to show sample macros](#) →

## Automatic macro recorder ▶ PLAY

If you don't need the full power (and learning hassle) of the scripting language, you can record simple macros pressing <F12> key to start the macro recorder (or use **Customize > Macros > Record macro** menu command). Use the mouse and the keyboard to do what you like (issue commands, tweak dialog controls and buttons, etc), then click on **finish** button to stop recording. The equivalent script representing your actions will be generated automatically and presented to you, just as if you had typed in the instructions. Then you can do slight modifications with the [macro editor](#).

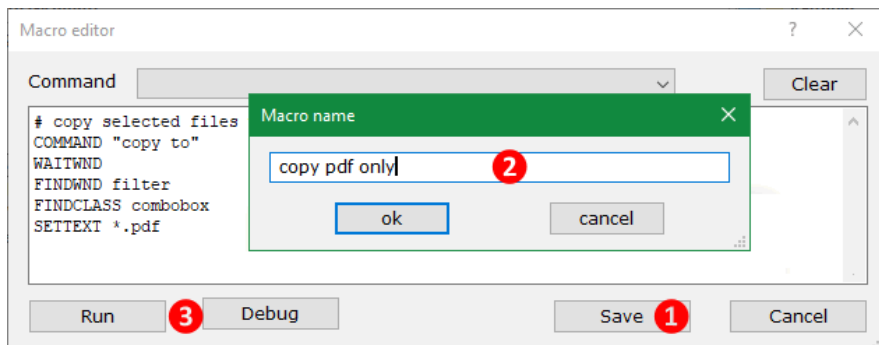


The recorder will only generate plain macros (without [loops](#)), and the resulting code looks spaghetti, but it should be adequate for the average macro requirements — and you don't need to do any "programming". If you start the recorder and click **Finish** immediately, without any actions, you will be presented with the last macro used.

## Saving and reusing macros

You can save useful macros for later in **Customize > Macros** list. The saving procedure is a bit unorthodox, but straightforward. First finalize the macro commands and test it well with the editor. When you are done, bring up the editor again (use **Edit macro** menu command to edit the last macro), click **Save** [1] button, give it a descriptive name [2] for the list, then **Run** [3] the macro. If you don't run it, it won't be saved. If you don't want to run it fully, you can "cheat" clicking the **Debug** button, then cancel the debugging without running anything. This way you will save the macro without running it.

The macro list menu works like [bookmarks](#), use its **Organize** menu command to add icons for toolbar and keyboard shortcuts for easy access. You can also *edit* an existing macro from organize dialog, adding or removing instructions, but it will feel a bit cramped. Another way to modify a saved macro is to first run it, then edit it (as the last macro used with **Edit macro** menu), and finally save it with the *same* name — which will overwrite the previous definition.



## Programmable column: file property algebra ▶ PLAY

You may be familiar with DeskRule's [desktop detective](#) panel that enables searching with combinations of file properties and attributes. xplorer<sup>2</sup> (**ultimate** edition only) programmable column is an extension to this concept: it allows you to combine file properties, boolean and arithmetic operators and a few functions into expressions, whose result is the value of the **Programmable** [S] stock column. This result can be then used as any other column, for display, sorting, filtering and searching for matching files. It is really versatile and the possible uses are too many to enumerate — just use your imagination!

To use the programmable column, switch to [detailed](#) view mode, right click on the column header row and pick **Programmable** from the menu. This will allow you to change the expression and will show the programmable column too:

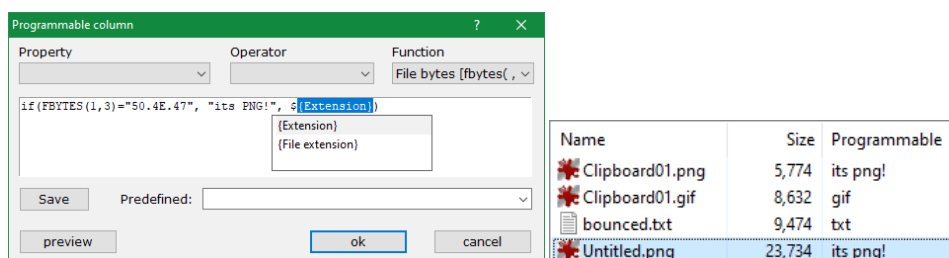


Figure 50. Define programmable column expression

The programmable column definition dialog has controls that guide you to create expressions, with drop-down lists for file **properties**, boolean and arithmetic **operators** and some simple **functions**. Or just type your expression in the edit box provided. As you type, it will **autocomplete** file property names for the more experienced users. Say you type \$EXT (the dollar is very important) a popup will find all the matching properties that have **ext** in their name and you can insert them easily — press the <Down> arrow key to select from the completion list.

All elements of programmable expressions are case *insensitive*

Click **preview** button to see the results of your expression on the selected file in the active folder pane. If something goes wrong, it will pop a syntax error with the likely error location.

All file properties ([columns](#)) can be used in the calculations. The rules can also search in files (GREP function) and extract specific file bytes (FBYTES function). The result of the expression can be either text or number or even a date, depending on the properties and operators used. If you use conjunctions of boolean operators, the result will be 1 or 0, for true or false respectively.

For instance the above example expression `if(FBYTES(1,3)=\"50.4E.47\", \"its PNG!\", ${Extension})` will check each file's *contents* to detect whether it is PNG format, in which case it will show *its PNG!*; for other file types it will show the extension. See how it works for the few files in figure 50 above.

Programmable expressions are *not* programs. The expression can be as long as you like but it is evaluated as a single line of "code". Think of cell expressions in MS Excel  
`=SUM(A1:A5)`

If programmable column is enabled, it applies the last expression you defined on each and every file, and shows the end result. There is only one programmable column, but you can save expressions you use frequently by supplying a name in the **Predefined** box in the above dialog and clicking **Save** button. This is similar to storing hyperfilters for search commands. Then expressions become available for future use from the history drop-down dialog control — pick one from the list and the programmable column display will be updated.

## File attribute algebra

This dialog allows you to do amazing feats which would normally require you to write a program or advanced shell script. You can combine any file property with arithmetic (add/subtract/multiply etc), relational (equal, greater than etc) and boolean operators (and/or etc) to create arbitrary expressions that can be evaluated for each file. For example take the expression:

```
(${Modified}) > ${Created} + 2) & ${Size} + 5KB < ${Size on disk}
```

It will match files whose modification date is later than the creation date + 2 days, *and* (&) their reported size plus 5 kilobytes is less than its total size on disk. As this uses the & (AND) boolean operator, the result will be 0 (false) or 1 (true). This value will show in the programmable column, and can be used as a [search rule](#) to identify files with custom qualities.

When you become familiar with the "official" property names and operators, you can type such expressions manually (press <Ctrl+DownArrow> key to autocomplete properties), or you can use the dialog drop-down lists to help you compose a complex rule with properties, operators and functions. A few clarifications are in order:

- **Variables.** These are file attributes, enclosed in braces with a leading dollar (\$) sign; `${Modified}` is the date last modified property (consistent with their use in \$-

tokens). The name inside the bracket is the same as the column name — it may include spaces. Type a \$ then start typing part of the property name and the dialog will offer you the possible name completions.



If you use special properties that not all files possess, the expression may fail. If for example you try to use "date taken" property on a text file, the expression will come up blanc (false always) — it is only supported for JPG files.

- **Constants.** Anything that is not a variable or a recognized operator is considered a constant. The type is automatically deciphered, e.g. 2 is a number, 3.4 is a floating point number, 25/12/15 is a date and so on. Numbers can be followed by KB/MB etc if you prefer size units.
- **Date constants.** Except for regular dates, you can use things like "yesterday" or "last year" (use quotation marks to preserve space characters). You can also use constants like "March 2015" which would match any date within the said month. For date ranges use the > and < operators in a conjunction e.g. `${modified} > "25 feb 2015" & ${modified} < "1 jun 2015"`
- **Wildcards and regular expressions.** A string constant can be either a literal like "hello world" or wildcard like `test.*`. Many of the special characters that appear in regular expressions could be mistaken for arithmetic operators, so please put wildcards and regular expressions in **single** quotes e.g. `'FILE[1-9].TXT'`. Also note that when comparing string operands, equal (=) implies identical strings (ignoring case), whereas the approximate (~) operator means that one string needs to be part of the other, e.g. a **partial** match.



Sometimes it is hard for the program to automatically fathom out a regular expression or differentiate it from a wildcard, if there are not enough special characters present. In that case, to force a regular expression enclose it all in brackets e.g. `'(a.+)'`. The meaning of the regexp will be the same and it will be unambiguously recognized as such.

- **Unusual operators.** Most operators have their usual sense, e.g. `1+2=3` will always be true. Here are a few exceptions:
  - When you subtract dates, the result is a number (their difference in days).
  - `$date + number` increases the date by that many days
  - Adding strings concatenates them `"str"+"ing" = "string"`
  - You can compare strings, e.g. `"a" < "box"` is true (like sorting strings)
  - All string comparisons are case insensitive so `"A"="a"` is true
  - Using brackets you can have really complex expressions with nesting.
  - Arithmetic operators are calculated first, followed by comparisons. Boolean operators execute last. Use brackets to override the default operator precedence.
  - Arithmetic between mixed types (except those mentioned above) will likely result in a syntax error
- **Helper functions.** Various string and date functions are supported like `len(text)` which returns the number of characters in the string argument (=4 in this case). Functions require one or more arguments that should be enclosed in brackets and separated with commas, e.g. `find(abc,b)`, and return values that can be used even within other functions, e.g. `mid(${Name}, find(${Name}, '12'), 5)` will extract a substring of 5 characters in the filename, after the first occurrence of 12 in said filename. Anything goes inside function arguments, including arithmetic operations etc. If you supply the wrong number or/and kind of argument to a function, a syntax error will be triggered. See the table below for all the supported functions.




**TIP:** If you want to use a string constant that happens to be a reserved function name, use quotes, e.g. `len("len")` is a valid expression that would return 3.

**Table 7.** Supported helper functions

Function	Description
<code>num=LEN(str)</code>	Return the number of characters in string, e.g. <code>len('abc')</code> returns 3
<code>substr=MID(str, nStart, nLen)</code>	Extract a substring out of a bigger text string, starting at position <code>nStart</code> and extracting <code>nLen</code> characters, for example <code>mid('abc12', 2, 3)</code> would return the string 'bc1'. Can be also used to access individual characters if <code>nLen</code> is 1.
<code>str=TRIM(str)</code>	Eliminate whitespace around a string, if any. It cuts off any leading or trailing spaces, e.g. <code>trim(' ab ')</code> would return 'ab'
<code>num= FIND(strHaystack, strNeedle)</code>	Find a needle in a haystack and return the position of the first occurrence — or 0 if nothing is found. <code>find(abc12,bc)</code> would return 2. The search isn't case sensitive.
<code>num=DPART(date,str)</code>	Access parts of the <i>date</i> , e.g. <code>dpart('march 2016', year)</code> would return 2016. The second argument <i>str</i> can return other parts of the date e.g. month or day (or even just d).
<code>date=DATE(str)</code>	Converts text to dates. Usually this is done automatically; still this function can be used without an argument ( <code>DATE("")</code> ) to return the current date and time. This can be used e.g. to calculate the file age in days.
<code>str=DATETIME(date,dfmt,tfmt)</code>	Format a date using separate date and time format strings. If you omit <i>tfmt</i> (3rd argument) no time will be shown. Format strings are the usual windows specifiers <code>yyyy MM dd</code> (see the documentation for <a href="#">GetDateFormat</a> ).
<code>xxx=MIN(xxx,xxx)</code>	Compare the two arguments and return the smaller one. This function accepts all kind of argument types (numbers, dates, etc) and returns the same kind. So <code>min(-1,2)</code> would return -1.
<code>xxx=MAX(xxx,xxx)</code>	Similar to MIN function, this returns the larger of the 2 arguments
<code>xxx=IF(term, if_true, if_false)</code>	Evaluates <i>term</i> and if it is true returns the second argument, else the third. The 3 arguments can be whatever makes sense for your conditional logic. <i>term</i> is true if it isn't empty or if it is non-zero.
<code>num=ABS(num)</code>	Returns the absolute value of a number, e.g. <code>abs(-1.3)</code> would be 1.3
<code>str=STR(xxx)</code>	Convert any type of argument into a string, useful for passing dates and numeric variables to other string functions. It can also convert hex strings obtained by FBYTES into plain text
<code>num=GREP(str)</code>	Search file <i>contents</i> for occurrence of text (can be a wildcard). If successful a positive value is returned, else 0 if keyword is not present in file.
<code>str=FBYTES(nStart,nLen)</code>	Extract <i>nLen</i> bytes from file, starting at file offset <i>nStart</i> (beginning=0). The returned string is the hexadecimal representation of the file bytes. For example PNG files can be identified because <code>FBYTES(1,3)="50.4E.47"</code> (use a hex file viewer to understand why)
<code>num=INT(xxx)</code>	Convert any type of argument into a number; it can also convert hex strings from FBYTES if up to 4 bytes long.
<code>substr=TOKEN(str, separ, nTok)</code>	Break up a string at <i>separ</i> points, and return the <i>nTok</i> entry (1=first token, -1=last). Special separator values are " " to break at spaces, \n which breaks at newlines and \t that splits on tabs. Similar in concept to CHOPSTR macro instruction. Example: <code>token("hello world", " ", 2)</code> returns 'world'.
<code>stdout=SYSTEM(strCmd)</code>	Execute DOS command <i>strCmd</i> and collect its console output. For example <code>system('dir "' + \${path} + \${name} + '"')</code> will execute the DIR command on each item (this example isn't particularly useful :). Note the use of 'single ears' to compose the command including the "double quotes" around the path name -- in case it contains spaces -- and the + operator used to compile the command string. With a little imagination it can be used as a poor man's (slow) column handler! Commands executed should be in %PATH% or full paths must be specified. External commands that don't write to a console (e.g. GUI programs) are not supported.
<code>num=DIRCNT(wildcard, type)</code>	Applies only to <b>folder</b> items and returns the count of direct subitems that match the <i>wildcard</i> ('*.JPG' will count all the JPG files, '*' will count everything; regular expressions allowed). The second argument can be <i>files</i> , <i>dirs</i> or <i>both</i> , to count items of the specified type ( <i>dir</i> =folder).
<code>xxx=FILE2(expr, path)</code>	Calculate a complex <i>expression</i> on a different file <i>path</i> for comparison purposes. The expression can contain anything (except for an embedded FILE2) and must be included in a single "quoted" string, e.g. <code>FILE2("GREG('hello world')", "d:\path\" + \${name})</code> . If you need quotes within the subexpression, use 'single ears'.
<code>xxx=FILEPROP(prop, path)</code>	Fetch a certain file <i>property</i> from a different file <i>path</i> , for comparison purposes. For example <code>\${Modified} &gt; FILEPROP(\${Modified}, "d:\path\" + \${name})</code> will compare the current item's date with its namesake in D:\path folder to see if it is newer. Note only a single property can be fetched, which should <b>not</b> be in "quotes" — but you can have multiple FILEPROP statements if required
<code>path=PNEXT(num)</code>	Returns the path of the next ( <i>num</i> =1) or previous ( <i>num</i> =-1) item, as they appear listed in a pane. This path can then be used in FILEPROP or FILE2 to fetch a property of a neighbouring item, e.g. <code>\${size} - FILEPROP(\${size}, PNEXT(1))</code> would calculate the size difference of 2 items (except for the last one in the list). PNEXT cannot be used for sorting or searching for items.
<code>ok=EXISTS(path)</code>	Test if a <i>path</i> exists. Return value is 1 if it is a file, -1 if a folder, or 0 if nothing exists.

If there is some problem with the complex expression like a misspelled file property or an operation between incompatible arguments, xplorer<sup>2</sup> will report a syntax error, highlighting the trouble spot. Correct the typo and you are ready to search.



Sometimes a property name on its own can be a useful rule, e.g. `${Keywords}` without any other operators will match files that have some tags defined. Or combine it with a leading NOT (!) operator to match files that do not have tags! Bare property names are also convenient for yes/no properties like `${IsFolder}`, otherwise you must be careful to compare them only against zero (=false or no).

The programmable column is useful for many advanced search tasks, e.g. find pictures whose width/height ratio is 4/3. It will also allow general tomfoolery like `(${Modified} - ${Date taken})/2 = ${Size}`, which arguably is not much use <g>. A final note is that extracting this column is rather **slow**; the more properties you use in an expression, the slower it gets. If you use multiple properties in a search expression, use the faster ones like filename and size **first**, and leave the slower ones (e.g. GREP) at the very end.


## User Interface customization

No 2 people are the same, so xplorer<sup>2</sup> is customizable to a minute degree, to make it your own — from basics as the window looks and layout, down to behavior of individual commands. Important tweaks are available from program options, others more specialized are available through an external settings editor.

Basic program [options](#) tweak how UI elements like the active folder view and the tree behave, whether you want dark mode interface and such. Using [layouts](#) you can define different workspaces, sets of panes and their positions and sizes to suit different file management tasks.

Under **Customize** main menu you will find several categories of favorite lists, that are maintained in a uniform fashion (Add/Organize/Execute as we saw for [bookmarks](#)), and in particular:

- **Toolbars.** Add up to 10 extra [toolbars](#) and customize each with command buttons for a particular task.
- **Column sets.** Store the [columns](#) of item information you want to see in detailed view mode.
- **User commands.** Store DOS or windows [commands](#) and programs you find executing frequently.
- **Folder groups.** Store related folders and reopen them in a [tabbed](#) pane, each folder in its own tab. This feature can be conceived as a multi-bookmark.
- **Custom groups.** Store item [categories](#) based on rules you define.
- **Macros.** Keep a library of useful UI automation [macros](#), which you execute often.
- **Layouts.** (this list is under **Window** menu) Remember the position and the visibility of view panes and toolbars. Storing different [layouts](#) you can quickly switch among different modes of operation

 When you add a new item in these lists e.g. with **Customize > Column sets > Add current** menu, you only define the item name. For further customizations, including a different icon and a keyboard shortcut, please use the **Organize** menu item of the list.

Each of these lists can hold up to 100 saved items, if you need more you must first delete some of the old unused entries. The same limit applies to saved [hyperfilters](#). If you want to see more items in Customize submenus, saving you the trouble of using **More...** menu commands, increase "Custom menu items" [advanced option](#). Then you will have direct access to more favorite customizations.

If you add any of these special favorite commands on a [toolbar](#), right click on the button and pick **Properties** menu command. This gives you a shortcut access to the list's **Organize** menu, where you can change the item's properties. For example you can edit the commands of a macro this way.

## Keyboard customization

Bookmarks and other items listed under **Customize** menu can be assigned custom keyboard shortcuts (in the range 0-9) using dialogs similar to [figure 8](#). To change the keyboard shortcuts of all other menu commands use **Customize > Keyboard** command. This way you can assign e.g. <F5> to be used for folder refresh and so on.

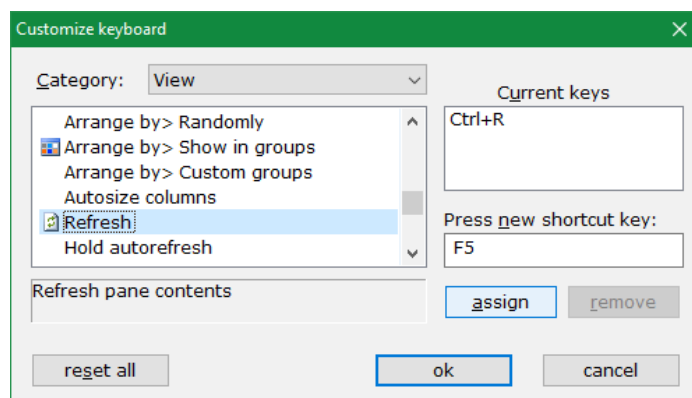




Figure 51. Keyboard customization dialog

The dialog lists commands organized by the main menu bar (*File, Goto, ...*). Use the **Category** drop down list to select the menu that holds the command you want to remap, e.g. for Refresh we need **View** menu. Scroll down to find the command you want and click on it. You will see its current keyboard shortcuts in **Current keys** box. To set a new key, click in **Press new shortcut key** box and type the key e.g. <F5>.

 If you try to set a key that is currently in use for a different command, the dialog will let you know about it; decide whether you want to keep the old command assignment or just remap it to the current command. Soon you must use complicated combinations with <CTRL> <SHIFT> and <ALT> keys because you will run out of simple accelerators.

Click **assign** button to set the new combination, it will be then listed in **Current keys** box. If you want to remove the old accelerator, select it and click **remove** button. To reset all key mappings to "factory defaults" click **reset all** button.

Change as many commands as you want, and click **ok** to finalize the new key bindings. All menus will be updated to show the new keyboard shortcut next to the command name. You only need to change command key bindings *once*, and next time you restart xplorer<sup>2</sup> it will remember your favorite keys.

 If you switch to **Miscellaneous** category, you will find many useful "hidden" commands that aren't in any menu. They are identified by number but you can see what they do in the command description pane (or use the [command finder](#) to search for obscure commands). For example command 57640 lets you *repeat* whatever menu command you used last — default shortcut key is <CTRL+Y>

**Customize > Keyboard use** menu command lists the current key assignments for all xplorer<sup>2</sup> commands. You can copy out the text, paste it to an editor and print them out so that you remember the quickest way to reach your favorite commands.

## Toolbar customization

As we have already seen, toolbars can be [moved](#) around, show large or small icons, and show command labels for clarity — or hide them to save space. You can also pick what buttons each toolbar contains. Right click on a toolbar and pick **Customize** from the menu:

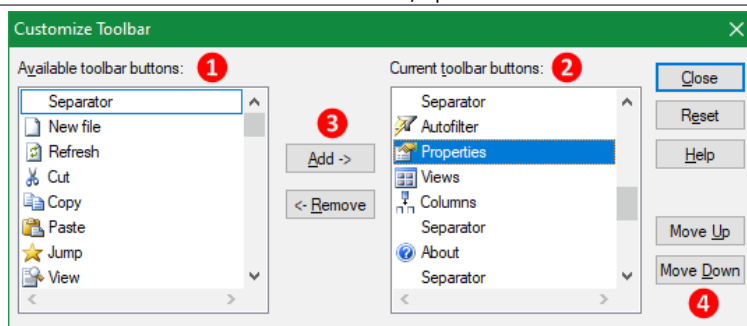



Figure 52. Toolbar customization dialog


Any command from the menu system that shows an **icon** next to it can be placed on a toolbar. You know which commands you need to use more, so you can place them on a toolbar for easy access. Also you can **remove** toolbar buttons that you don't use that match and replace them with others you enjoy more.

The dialog shows all buttons that are **Available** [1] (all menu commands with icons), and a list of buttons **Current** [2] on the toolbar. Select any command from the Available list and click on **Add** [3] button to add it to your toolbar; likewise select a button from Current list and click **Remove** to get rid of it.

Finding a command to add can be a bit of an ordeal, as there is no way to search for commands, you just scroll the list till you see the icon you are after. Toolbar button names may or may not be similar to their respective menu commands, sorry about that! As for commands in **Customize** menu (macros, user commands, bookmarks etc), you can identify them by their list name.

 If you are looking, say, for the user command you added with the label **WINDIFF**, click in the **Available** list [1] and press the **first letter W** a few times. It will show you buttons that start with W and you will soon locate the one you are after!


The current list shows items in the order they appear on your toolbar. Select an existing command and move it up/down with the buttons [4]. When you add a new button, it gets inserted **before** whichever item in the current list you have selected. If the new button ends up somewhere you don't like, use move up/down to relocate it.

 You can reorder toolbar buttons with the mouse, when you don't see the customization dialog. For a normal toolbar hold down **<SHIFT>** key and click on a button. You will see a big rectangle drawn around the button, then move it with the mouse and drop it to its new position. Drop it outside the toolbar to remove it completely!

Add separator buttons (the first one in **Available** list) to organize toolbar buttons in groups. These are the little vertical bars you see. You can add as many separators you need and position them to split groups.

When you are done customizing buttons, click **Close** button to see your new toolbar. Or if you changed your mind, click **Reset** button to undo your recent changes.

Use **Customize > Toolbars > Add new** menu to add a new toolbar with a name, and get to customize it immediately, adding buttons for a particular task. You can have up to 10 such extra toolbars. It makes sense to organize extra toolbars in a task-oriented fashion, bunching relevant buttons together. Whenever the need arises you activate the toolbar using **View > Toolbars** menu (or right click on any toolbar). You don't have to keep all toolbars visible at all times. Toolbar visibility is one of the properties associated with **layouts** — each layout can have its own set of active toolbars.

 You can do away with the limitation about icons and put **any** menu command on a toolbar through a macro. So you create a simple macro that just invokes the menu command, add it to the customized list, and then it has an icon and can be added on the toolbar indirectly! For more details see this [blog post](#).

## Toolbar skins

For a quick face-lift of xplorer<sup>2</sup> appearance, you can download and install custom toolbar and menu item images that other xplorer<sup>2</sup> users have created and shared with the community. To see what's available [click here](#).

You can also create your own skin. It's **easy** and no programming is required. Just grab the original button images and modify to taste with any image editor, then drop them in the installation folder. You may also consider sharing them with the rest of the world - just drop us an email.

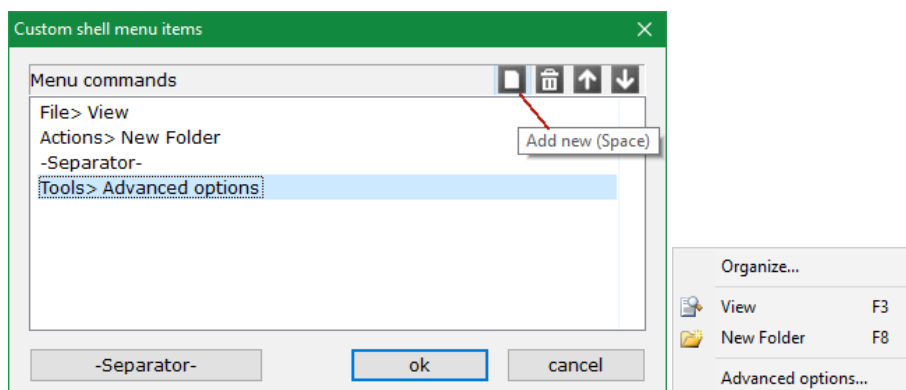
Download [skin designer pack](#) (17 KB)

Nowadays you need two skins, for the 2 available toolbar icon sizes. One should have buttons 16-20 pixels square (x2skin.png), and the other should be any size you like up to 48x48 to match your screen's DPI (x2skin\_XL.png)

## xplorer<sup>2</sup> context menu

The shell context menu is a system menu that contains commands to handle selected files and folders, simple things like rename and properties, plus any advanced commands you added by way of context menu shell extensions. xplorer<sup>2</sup> adds a couple of its own menu commands at the top of this context menu, under xplorer<sup>2</sup> heading.

You can put any of xplorer<sup>2</sup> menu commands in this submenu, whatever you need to have handy as a context menu command. Use **xplorer<sup>2</sup> > Organize** command to add and remove items from this special menu:

Figure 53. Organize xplorer<sup>2</sup> submenu

Use the little toolbar to add, remove and reorder existing menu commands. When you click **Add new** you get a dialog similar to [keyboard customization](#), where you can select a menu command by category as they appear on the main menu bar. You can add **separators** too for grouping commands. When you are all done click **ok** button to finalize the menu; commands will appear nicely with icons and keyboard shortcut information, see the above picture.

This submenu is inserted in the context menu of all items, active folder, folder tree, even the background menu where you click "nowhere". If you don't want to clutter your shell context menu with extra commands, you can remove xplorer<sup>2</sup> menu using the [advanced option](#) "Don't add xplorer<sup>2</sup> commands in shell context menu".



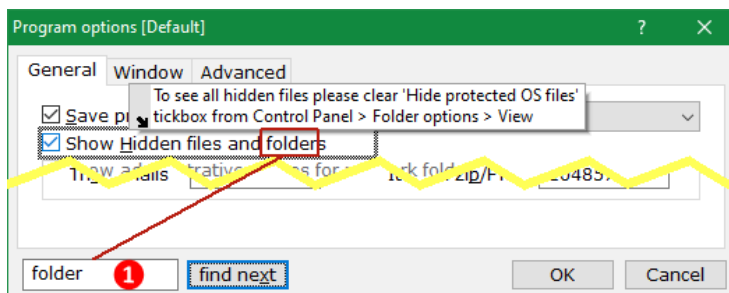
## Program options

xplorer<sup>2</sup> has many options that you can fine tune so that it behaves to your liking. Use Tools > Options menu to bring up the program options dialog, a property sheet as in the picture to the right with basic tweaks. There are even finer options to tweak, using **Tools > Advanced options** menu command, which opens a separate program with a plethora of possible adjustments.

The main options dialog is organized in three property pages, each focusing on a section of the user interface. Most options are self-explanatory once you read the checkbox description. Many options come with a help **tooltip**, which pops up when you leave the mouse pointer hanging over the option, helping you understand what they are meant for.

If you have difficulty locating an option, use the search box [1] at the bottom left corner with a suitable keyword. Click on **Find next** button a few times till you locate the option you seek. Each matching option appears **highlighted** with a thick border.

Most program options are specific to the current **layout**. You see the layout name on the option dialog's **titlebar** (it is the [Default] layout in this example).

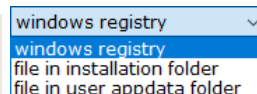


A fundamental option is "Save program state on exit", which controls the persistence of all other settings. If you have this option ticked, xplorer<sup>2</sup> will **save settings** on exit, meaning the state of the window, all the tabs and any recent changes to program options, will be remembered; next time you run xplorer<sup>2</sup> it will look exactly as you last left it. Turning this option off isn't recommended; if you want xplorer<sup>2</sup> to start at fixed locations use [command line](#) arguments instead of disabling saving settings on exit.

Changes you make to global properties like bookmarks and items in **Customize** menu are *always* saved on exit, regardless of this option. If you have it disabled, use **Actions > Save settings now** menu command whenever you want to save a snapshot of your latest settings.



xplorer<sup>2</sup> professional saves settings **only** in the windows registry. This is the recommended location even for ultimate edition, when you have it installed in your home/base computer. If you want to run ultimate edition in [portable](#) mode, pick "File in installation folder" from the drop-down list as the destination for saved settings. This will create an INI file in the flash drive, next to the program.



You can change as many options as you like, switching property pages as necessary. When you click **ok** button the changes you made will take effect immediately — well most of the time! There are some major options like turning dark mode on/off that necessitate opening a *new window* — this is done automatically with minimal interruption to your workflow.



When you quit xplorer<sup>2</sup>, only the active folder tab will save its settings. When you restart xplorer<sup>2</sup>, all tabs will look **identical**. If you want to save settings for each tab individually, tick the [advanced option](#) "Each folder tab can have different settings"

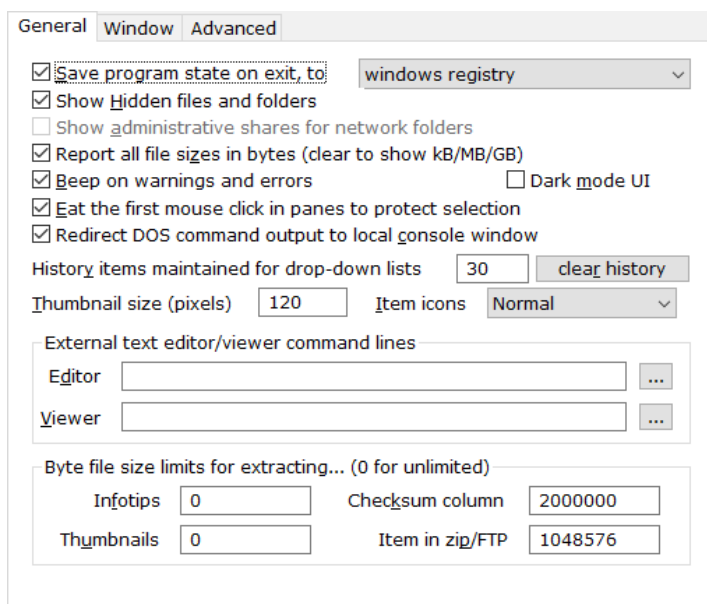
### Further reading

If you have problems with settings not persisting properly, check the options [troubleshooter](#)

## General options

The first property page has general options, the most important of which are:

- **Show administrative shares.** If you access network PCs you normally only browse public shares. This option (requires elevation) shows administrative shares like C\$ that give you local filesystem hard disk access to remote PCs
- **Dark mode UI.** Tick to enable the fashionable dark mode where everything is on black background. You don't get much flexibility choosing colors in this mode. Untick the option to return to normal window colors.
- **Eat the first mouse click.** If you click to activate a folder pane with your mouse where many items are selected, you risk losing the selection, unless this option is ticked. Alternatively press <TAB> key to switch folder activation.
- **Redirect DOS command output.** Tick to use the modern xplorer<sup>2</sup> console [replacement](#), otherwise a normal MSDOS console window is used for external commands.
- **History items maintained.** How many past inputs to keep for combo boxes. Each dialog has 1-2 comboboxes where you enter your text parameters; this number controls how many are remembered. If you type more than this limit, the older ones are discarded. History items assist repetitive commands with [autocomplete](#). Privacy concerned users should click **clear history** button and set history items to 2, which means remember nothing.
- **Thumbnail size.** Basic pixel size for thumbnail view mode. This also affects the size in **Mini thumbs** view mode, through the [advanced option](#) multiplier "Mini-thumb size factor". You can see bigger thumbnails with dynamic zoom (<CTRL> + mousewheel)
- **Item icons.** Select what icons to show for files and folders using the drop-down list. The available options are
  - **normal** for rich regular icons and overlays
  - **simple** for faster document-related icons; won't show icons for executables
  - **plain** for super fast plain icons (just 3 types); will show only basic icon overlays
  - **no icons** will show plain text listings, very frugal. Thumbnail view modes work
- **External text editor/viewer.** If you don't like the internal [editor](#) for text documents, enter here your alternative full paths for the replacement. You should leave the **Viewer** blank though because editor<sup>2</sup> reads search keywords from xplorer<sup>2</sup> content search commands, and you can find them with <F3> once the text file is loaded.
- **Byte size limits for extracting.** Some legacy file size limits for certain operations. Set them to 0 to cancel the limit. You should mainly worry about "Item in ZIP/FTP" that is the maximum size that will be extracted from [archives](#) for searching and "Checksum column" that is the biggest file to calculate MD5 checksums (they are quite slow accessing the entire file)



## Window options

The second property page has options for folder [view](#) panes, where you browse folder contents, the folder [tree](#) pane and the [quick viewer](#), each in its own group of checkboxes.

The most important settings are:

- **Single click activation.** Click once to open items — instead of double click
- **Infotips for items.** Shows popup details when you leave the mouse button on a file; the information is file type related. In scrap windows it also shows the full path.
- **Hover selection.** Oddball selection mode by hovering on an item. Not really suitable building large selections — incompatible to sticky selection mode.
- **Automatically re-sort contents.** When autorefresh senses a change in the folder contents, any new files are inserted in their rightful sort position — otherwise they are added last. Causes panes to "jump" in fast changing folders like Downloads.
- **Highlight primary sort column.** The sort column in detailed view mode shows with a different background color.
- **Mouse peek preview.** Turn [peek preview](#) with both mouse buttons on/off.
- **Folders sorted.** Using the drop down list you control whether folder items are sorted mixed **as files**, or **separately** (first of all) or **after files**.
- **Hands free activation.** Load folders in the active pane as you browse the tree with arrow keys, without pressing <ENTER> — after a small timeout.
- **Automatic branch expansion.** During drag-drop, if you hover your mouse over a tree node it will expand so you can locate subfolders.
- **Keep synchronized with folder.** The tree aligns with the active folder, as you browse, change tabs etc. Otherwise you can manually align the tree pressing <ALT+T>
- **Show 2 trees.** Shows a separate tree for each left/right pane. Hides the quick access list too.
- **Text preview size.** Only a few bytes from a text file are loaded in the previewer pane in draft mode. Increase this number to see more file content, but expect delays.
- **Start playing media files immediately.** Plays music and movies automatically when you select one — otherwise you must click on play button. To enable this option play something first.
- **No hex preview.** Tick to disable the hexadecimal preview of files that don't have a normal preview (e.g. to stop seeing binary contents of EXE files).
- **Main font.** Choose a bigger or smaller font for folder contents and other window panes. Fonts for menus, titlebar etc are controlled by your system display settings
- **Inactive background.** Choose a different color for the background of inactive panes. Default is a slightly darker tone than the active folder pane — which is white.
- **Plain menus.** Tick to hide less frequently used commands for a less confusing menu bar. Untick to show all menu items.
- **Dialog balloon help.** Clear this box if you don't want to see the explanatory [balloon](#) help tips in dialogs.

Advanced options

The third property page has more advanced behavior options, the most important of which are:

GeneralWindowAdvanced

View panes

☐Single click activation

☐Grid lines in detailed mode

☐Infotips for items

☒Full row selection

☒Allow slow double-click to rename items

☐Hover selection

☒Automatically re-sort contents after moving/copying etc.

☐Highlight primary sort column

Max list column width

196

Folders sorted

separately

Dual pane alignment (tiling):

☐Horizontal

☒Vertical

Tree

☒Single click to change folder & "hands-free" activation

☐Automatically shift focus to view after activation

☒Automatic branch expansion when hovering on a node

☐Keep synchronized with folder in active view pane

☒Show quick access list

☐Show 2 trees

Quick viewer

Text preview size (bytes)

4096

Tab width

3

☒Start playing media files immediately

☐No hex preview

Main font...

Inactive background

☐Plain menus

☒Dialog balloon help

- **Status bar panes.** Tick to enable or disable certain [statusbar](#) information panels.
- **Highlight active pane's titlebar.** Change the active pane's titlebar color so you can tell it apart easily, not just by background color. This is recommended if you use dark mode UI.
- **Extract custom icon overlays.** Enable icon overlays through shell extensions, eg. for TSVN repository controlled folders. Otherwise only show basic overlays for shortcuts and shares.
- **Quick preview docked to the right.** The free [lite](#) version doesn't support [docking](#) panels; tick this to put preview to the right side of the window — otherwise it goes under the tree.
- **Explorer-style path autocompletion.** Autocomplete paths like windows explorer, using a popup window. Otherwise autocomplete path stubs pressing <F1> key repeatedly till you find the correct folder.
- **Close old window after cloning.** Tick to close the old window when you clone a new layout window, otherwise both remain open
- **Preserve modified date for comments.** Setting a file [comment](#) will not change its modified date — if you so prefer
- **Faster network access.** Tick for faster network access albeit a bit poorer on visuals. Disables posh icons, [folder size](#) calculation and automatic tree alignment.
- **In-place edit of file properties.** Turn on the convenient [spreadsheet](#) property edit mode
- **Folder options.** Your windows folder options are largely irrelevant to the behavior of xplorer<sup>2</sup> but there are a few that do matter, like hiding extensions and operating system files. This button gives you access to your control panel
- **Undo explorer replacement.** If you change your mind about xplorer<sup>2</sup> [replacing](#) windows explorer, use this button to restore windows explorer. It will probably ask you to run xplorer<sup>2</sup> elevated to complete the system wide modification.

GeneralWindowAdvanced

Status bar panes

☐ Show total folder size

☒ Show disk free space

☒ Show active item information

☐ Show item index

☐ Folder tabs on top

☐ Highlight active pane's titlebar as in 2xExplorer

☒ Natural number sort (XP or later)

☐ Don't lock browsed folder

☐ Calculate subfolders size automatically

☒ Extract custom icon overlays (e.g. for CVS)

☐ Quick preview pane docked to the right side

☐ Explorer-style path autocompletion in addressbar

☒ User friendly date column information

☐ Close old window after cloning a new layout

☒ Show column header for all view styles (Vista or later)

☒ Automatically resize Name column in detailed view

☐ Preserve modified date when setting file comments etc

☒ Faster network access with plain icons etc

☒ In-place edit of file properties in detailed view mode

folder optionsundo explorer replacement

Advanced options tool

xplorer<sup>2</sup> offers many more micro-adjustments to its working with registry editing (see [details](#)). As registry isn't everybody's cup of tea, there is a user friendly tool to help you with these micro tweaks. Use **Tools > Advanced options** menu that will quit xplorer<sup>2</sup> and start the settings editor:

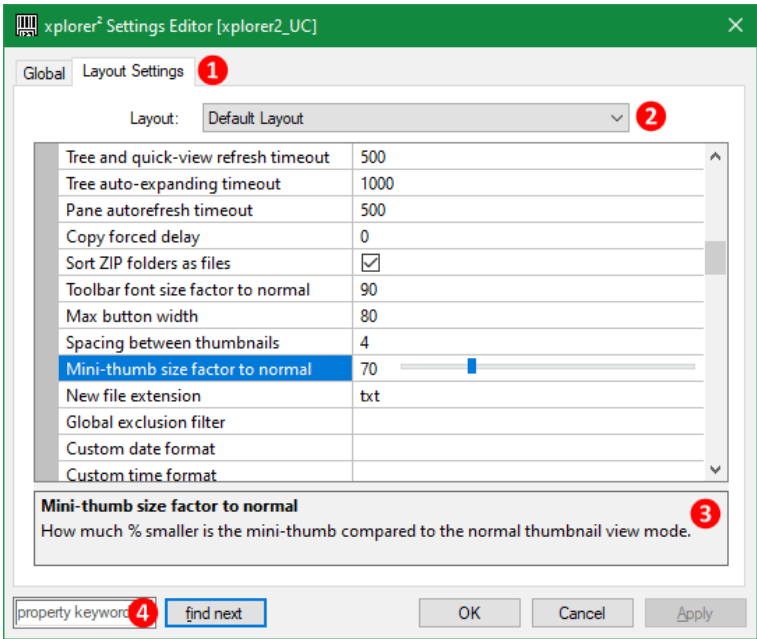



Figure 54. Advanced options tool

Advanced options are split in two categories, **global** and **layout**-specific, and are selectable using the tab control [1]. Global options affect all layouts, and layout options are just for a single [layout](#). Use the layout selector drop-down list [2] to choose which layout you want to change. The layout selector is set to the xplorer<sup>2</sup> layout that was active when you started the external settings tool

There are too many options to describe in detail, the most important ones were mentioned in the body of this document, so you can **search** for them by name using the search box [4]. Or use a generic keyword like *thumb* and click **Find next** button a few times to see what tweaks are available related to thumbnails.

 Searching for options works in the current tab only, find next will cycle through the matching options in the current tab. Switch to the other tab (global or layout [1]) to search all possible matches for your keyword. If a property isn't found in the active page, the tab will be switched automatically for you.

The main body is the property control that lists all tweakable program options, and lets you change them. There are numeric, text, color and other types of properties. As you select properties, you get to read brief explanations about what each one does in box [3].


After you are done changing options, click on **ok** button to save them and then restart your xplorer<sup>2</sup> to see them in action.

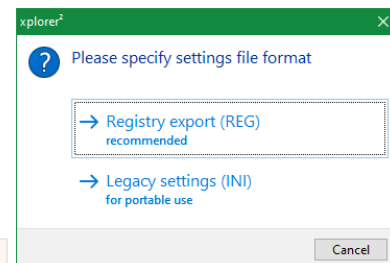
## Backup and transfer settings

You spend time and effort to tweak xplorer<sup>2</sup> settings to your liking, so you want to safeguard them against the numerous setbacks that are unavoidable with computing. Hence you should make a **backup** of your settings every once in a while, and keep them safe in case you need to restore them.


xplorer<sup>2</sup> saves its settings in the registry (or INI file if you run ultimate edition in portable mode). Use **Actions > Export settings** menu to save your current settings to a standard registry information REG file. Ultimate edition gives you an option to create an INI settings file, see the question to the right. All settings are exported, window sizes, dockable panes, fonts, tabs, layout information etc. Keep this x2settings.REG file (or however you chose to call it) somewhere safe, perhaps in your [backup](#) disk drive.

If a case arises where you need to **restore** settings, one possibility is to fully quit xplorer<sup>2</sup> with **File > Exit** menu, then double click on the previously exported x2settings.REG file which will be automatically imported. When you restart xplorer<sup>2</sup> it will look like it did when you created the settings snapshot. Another way is to do **Actions > Import settings** inside xplorer<sup>2</sup>.

 You must browse the REG file you want to import and **select** it *before* using the import menu command. You are effectively using xplorer<sup>2</sup> as a giant Open dialog. For ultimate version, ensure your current settings persistence [option](#) matches what you are trying to import (must be *registry* for REG and *file* for INI import).



Another use for backed-up settings is for troubleshooting problems. After series of tweaks you may find your xplorer<sup>2</sup> is in a rut and won't work properly. An extreme attempt to bring order is to fully uninstall then reinstall xplorer<sup>2</sup> with its "factory" settings. Export your settings *before* such drastic measures so you can restore them — but chances are the problem will return.

 Settings are organized in registry keys. There is one key for global options, one for the default layout and then any number of keys for other layouts you have defined in **Window** menu. So the following registry keys are in use:

```
HKCU\SOFTWARE\ZabaraKatranemia Plc\xplorer2_UC
HKCU\SOFTWARE\ZabaraKatranemia Plc\xplorer2_UC.global
HKCU\SOFTWARE\ZabaraKatranemia Plc\xplorer2_UC.customLayout
... (other layouts)
```

Finally if you want to move xplorer<sup>2</sup> to a new PC, export its settings from the old one (before uninstalling), then import them to the new PC. Not all settings are transferrable, e.g. bookmarks and user commands with full paths expect a similar folder hierarchy organization — which may be absent (even small path differences make bookmarks unreachable).

### Further reading

- [Migrate](#) your license and settings to a different PC
- Registry hack to [reset settings](#) without uninstalling — and with minimum harm to your basic xplorer<sup>2</sup> settings

## Window layouts

One moment you browse general files and need a dual pane generic file management window, the next you are in a folder full of photos and you need a large previewer pane to examine them. You could use **View** menu to turn panes on and off to suit your current browsing needs, but for large UI changeovers you can save and reuse layouts (aka workspaces) that let you change the window appearance in one stroke.

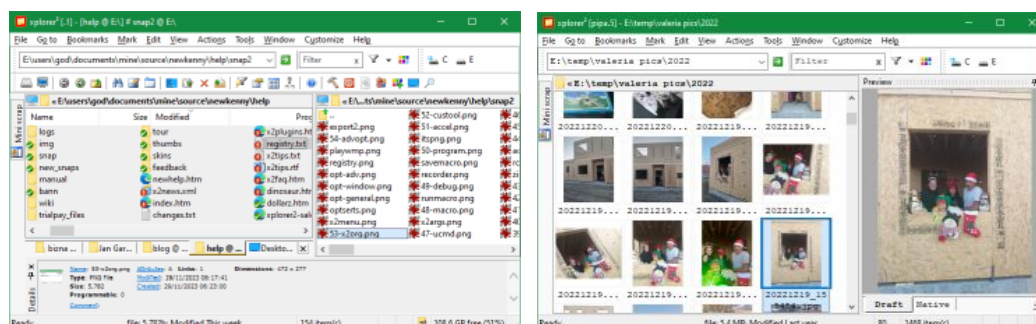




Figure 55. Two sample layouts side by side

A layout keeps information about the entire xplorer<sup>2</sup> window and in particular:

- Whether it is [ribbon](#) or menu-based
- How big is the window and what panes are visible (tree, quickviewer etc)
- How big is the main UI font
- How many toolbars show and what command buttons do they contain
- How many tabs appear and which folders you browse in them
- Is it dual or single pane, dark mode or white?
- The [mini-scrap](#) pane contents may differ (if pane is turned on)
- Most [options](#) are also layout-specific, e.g. you can have single click activation only in one layout and double-click for the rest.

xplorer<sup>2</sup> starts in its default layout. You should first do all your customizations to make sure this basic layout meets your most frequent demands. For some people that's all that's required. If you want to define special workspaces on the other hand, make your changes to the window and its options, then use **Window > Save layout** menu to define a new layout with a descriptive name for your reference. Then use **Window** menu to switch among the layouts you have defined, or back to the **Default** layout. Whenever you choose a layout from **Window** menu, a new xplorer<sup>2</sup> window opens for it. You can read a custom layout name on the window's titlebar in [square brackets]:


 xplorer<sup>2</sup> [x2solo.2] -

 When you first save a layout make sure you give it both a name and optionally a longer **description** about what it is meant for. This is the only time you get to set a layout description — without having to resort to registry editing. This description shows on the [statusbar](#) as you traverse the layout items list in **Window** menu.

The bottom half of **Window** menu is a list organizing saved layouts, like we have seen many times in **Customize** menu and [bookmarks](#). Use **Window > Organize** to assign icons and shortcut keys to saved layouts — note you cannot rename or reorder them like other favorite lists but other than that the list management is similar, e.g. you can delete unwanted layouts.

These limitations are due to the fact that a layout is an entire registry key with your custom window settings

Some people use layouts to organize different sets of folder tabs, but that's not the recommended use case. If the only thing you need is to switch folder tabs, you can use [folder groups](#), perhaps combined with a custom desktop shortcut to launch xplorer<sup>2</sup> with these particular folder groups using [arguments](#). At any rate each layout starts with a fixed set of folders every time (unless you ticked "Disable per-tab setting policy..." [advanced option](#)).

 Each layout is a dynamic changing entity. As you use the layout, change folders, close tabs, change pane styles, all these modifications are persisted and modify the essence of a layout. When you re-clone it, you will start where you left it last time. If you want a layout to be **frozen** and always stay the same, tick off "Save program state on exit" option — which is itself tied to a particular layout!

When you start xplorer<sup>2</sup>, it comes up in its default layout. If you want to start with a different layout, use `"/S:layout name"` command line argument. However it is recommended you make your basic modifications to the **default** layout so you don't need to invoke another one as your default.

---

◀ [Customization](#) | [Contents](#) | [Appendix](#) ▶

## Installation and registration

xplorer<sup>2</sup> is designed for Microsoft Windows operating system (albeit some run it on Linux and Mac using emulators like Wine and Parallels). Once upon a time it run on everything including windows 98, but nowadays the minimum platform is windows 2000 — but it is tested only for windows XP and later, including the latest windows 11 version.

You can download the **free trial** from the [website](#) and evaluate it for free and full power for a period up to a month, then you must either buy or uninstall it as it will stop working. At any point you can **uninstall** xplorer<sup>2</sup> using **Add or remove programs** applet from your system settings (or control panel)



**Designed for:** Windows XP and later

**Languages supported:** English, French, German, Japanese, Spanish, Chinese, [more](#)

**Free trial period:** 21 days

xplorer<sup>2</sup> comes in 3 editions, a **free** lite version that has basic functionality, and the paid **professional** and **ultimate** editions. Commands missing from the free lite version are clearly marked with **[P]** in the menus, and will pop an advertisement for professional version if selected **[P] Scrap container**

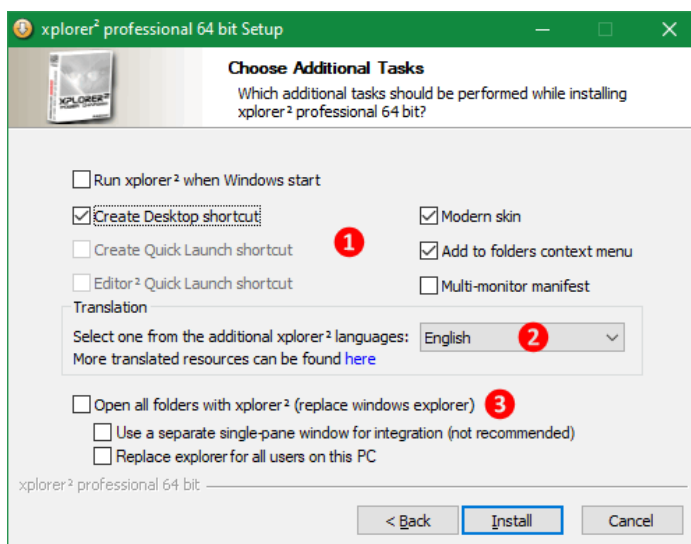
Normally the [download page](#) detects your operating system version (usually 64 bit windows nowadays) and lets you have the correct installer for your OS. Sometimes when your important [shell extensions](#) are 32 bit, you may want to download the 32 bit xplorer<sup>2</sup> installer that integrates with 32 bit DLLs, using the version selector next to the download buttons (see pic to the right).

Click on the download button for the edition you are after (pro or ultimate) and once it is finished you can run the installer. It is digitally signed and virus free so there shouldn't be any problems running it.



The installer is standard, you get to pick the installation location etc. When you reach **Additional tasks** page (see picture), you can choose how to install xplorer<sup>2</sup>. There are some general option checkboxes **[1]**, a selector for GUI translation **[2]** if English isn't your cup of tea, and an option to replace windows explorer **[3]**. Most options are self-evident, here are brief descriptions:

- **Create desktop shortcut.** Puts xplorer<sup>2</sup> shortcut icon on your desktop so you can start the program easily. You can also pin it on your taskbar or use **Start** menu to type the program name
- **Modern skin.** Includes nice and big toolbar buttons, recommended to leave it enabled
- **Add to folders context menu.** Whether to add a "Open with xplorer<sup>2</sup>" command in shell context menu for folders, lets you start xplorer<sup>2</sup> with a particular folder easily.
- **Multi-monitor manifest.** Tick this if you are using many monitors with different DPI resolution, it will make xplorer<sup>2</sup> automatically adjust its window when you move it across monitors, changing fonts etc so it remains scaled and readable. If this doesn't work well for you, run the installer again and leave this box unticked — and try to sort things out with compatibility [tweaks](#)
- **Translation.** **[P]** **PLAY** Pick one of the available languages using the drop-down list. Note that translations are only for the user interface; all help and support are offered in **English**. You can switch between a custom translation and english using **Help > English translation** menu command. Translations are maintained by volunteers; if you find any problems with the quality please contact support.
- **Replace windows explorer.** Tick to use xplorer<sup>2</sup> to open folders by default instead of windows explorer. Note this option does less than most people expect it to do. Windows explorer still exists and runs your desktop. When you double click on folder icons on your desktop, xplorer<sup>2</sup> appears, that's all. You may as well run xplorer<sup>2</sup> and open all your folders within the program. Some programs that launch folders do not respect this association and open windows explorer directly. System open/save dialogs cannot be replaced; to help navigate these dialogs copy [folder paths](#) from xplorer<sup>2</sup>. To run windows explorer, right click on a folder and tick **Open** from the context menu (not **Open with xplorer<sup>2</sup>**)
- **Replace explorer for all users.** Tick to replace windows explorer for all users — or just for you. It may upset your spouse if she shares the same PC!



When you tick the option to replace windows explorer, xplorer<sup>2</sup> automatically starts whenever you log in. It also takes over <WIN+E> key to open itself. When you close xplorer<sup>2</sup> window, the program stays in memory (you will see it in program manager in a low resource state); to fully quit xplorer<sup>2</sup> use **File > Exit** menu command. If you have problems with folder windows opening out of the blue, disable the explorer replacement mode by uninstalling then reinstalling xplorer<sup>2</sup> without ticking this option **[3]**

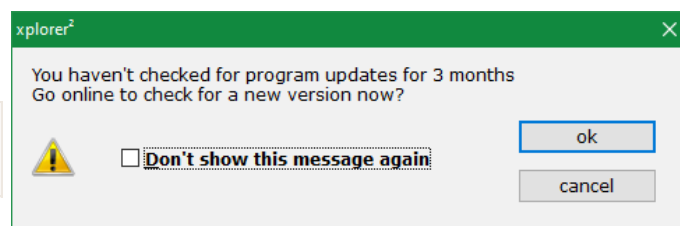
### Program updates

xplorer<sup>2</sup> is under constant development, and updates are released a couple of times each year with added features and bugfixes. Every 3 years or so there is a big [paid for](#) update where the major version number changes e.g. from 4.x to 5.x. Minor updates are free.

Every few months or so xplorer<sup>2</sup> will prompt you to check if there's a newer version available. If your license allows it, download and install the latest and greatest version. To cancel these reminders tick **Don't show this message again** dialog box. From then onwards you can manually use **Help > Check for updates** menu command to check for newer versions.



When you install an update, it's best to install it **over** the old version, so that your settings and license key will be preserved. If you want to do a clean install on the other hand, uninstall first and answer **YES** to both questions (remove settings and license file).



Use **Help > What's new?** menu command to read what features were added in the latest version.

### Ultimate (portable) edition


Ultimate edition is the best xplorer<sup>2</sup> money can buy. It is more expensive but you get the best algorithms for search, advanced features like the [programmable](#) column and portability. See this [comparison table](#):



Table 8. Comparison of xplorer<sup>2</sup> editions

	Professional	Ultimate
Explore and preview	✓	✓
Folder tabs and dual panes	✓	✓
Advanced search & file management	✓	✓
Fast desktop search	—	✓
Portable use (without installation)	—	✓
Unique <a href="#">extra</a> features	—	✓


If you are using xplorer<sup>2</sup> on the move you will find that saving program settings in the windows registry is not an option. One of the main advantages of xplorer<sup>2</sup> **ultimate version** is its portability. Using [options](#) ("Save program state" drop-down list) you can tell xplorer<sup>2</sup> to store its settings in an INI file instead of the registry.


 If you keep your ultimate version on a USB stick then pick the settings persistence option "file in installation folder", which will save the settings in a file called X2SETTINGS.INI next to the xplorer<sup>2</sup> executable program. If you installed the ultimate version on your hard disk and still don't want to use the registry, pick the option "file in user appdata folder" (usually C:\Users\<you>\AppData\Roaming\zabkat), as the installation folder inside C:\Program Files will most likely be protected and inaccessible. INI files are slower so if you have xplorer<sup>2</sup> installed on your home base, you'd better save settings in the **registry**.

If your **company** regulations won't allow you to install programs, you can create a portable xplorer<sup>2</sup> installation on a USB stick (flash drive) and run it from there without installation — assuming you are allowed you to run stuff off USB sticks at work.

Here are the **step-by-step instructions** to place xplorer<sup>2</sup> ultimate in a USB stick environment. For example assuming we want to setup the portable version within F:\x2portable\ folder:

1. Install xplorer<sup>2</sup> ultimate on your home computer (on your C:\ hard disk) and use your unlock key to activate it.
2. Setup the program options to taste.
3. Use the menu command **Actions > Export settings** to write the settings to the file F:\x2portable\x2settings.INI — click on [legacy settings](#) task button
4. Copy all the files in the installation folder (typically C:\program files\zabkat\xplorer2) *including* the license file X2.LIC to the USB stick (F:\x2portable)
5. (optional) If you are using the [mini-scrap](#) pane make sure you place the CIDA contents file on your USB stick too, by right-clicking on the miniscrap and choosing **Save as** command - but note that all contents must be on the USB stick.

If you have problems setting this up please watch this demo video: 

 Not all your favorite settings may be reusable in different computers; e.g. some of your bookmarks may point to inexistent folders. Using environmental variables in path definitions may alleviate this problem. If you want to have portable [user commands](#), install your tools in subfolders of your xplorer<sup>2</sup> portable location (under F:\x2portable\ ) and use %X2DIR% environmental variable to address them instead of fixed paths — it points to xplorer<sup>2</sup> installation folder. For example edit the user command description to %X2DIR%\tools\myTool.exe which will be equivalent to F:\x2portable\tools\myTool.exe but won't depend on the drive letter your USB stick happens to be assigned with.

After you properly set up your portable edition in a removable medium like a USB stick, then you have these advantages:

- **No installation** is required to be used in any computer. Even if you don't have administrative privileges you can enjoy your xplorer<sup>2</sup> wherever your work happens to bring you.
- No traces left in the host computer. After you quit xplorer<sup>2</sup> the foreign registry will be unmodified.

To ease the way you access xplorer<sup>2</sup> from your removable media you can create an AUTORUN.INF file in your root USB drive folder, and set it to open xplorer<sup>2</sup> (whenever you plug in the USB stick), editing it as such:

```
[autorun]
open=x2portable\xplorer2_64.exe [optional arguments]
```

## Program registration

xplorer<sup>2</sup> is a "try before you buy" application. You can download and install it, then use it for free at full capacity for a limited time period. This way you can tell if it is the right tool for you before buying it. The longer you take to evaluate the more nags it will throw at you, including this nag dialog when you start it:

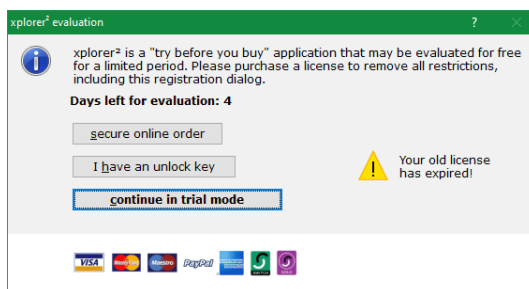



Figure 56. Registration prompt nag dialog

Click on **Continue in trial mode** to proceed without a key. Eventually after a period of evaluation at full capacity, it will ask for an unlock key to let you run it. At this point you should make up your mind, whether to purchase or uninstall. Use **Secure online order** to [buy a license](#) — all major credit cards and paypal are accepted for payment.

xplorer<sup>2</sup> is licensed **per user**. If you are the only user and have 2 computers, you only need one license key. But you must add a seat for each extra user. Your purchase will entitle you to free upgrades and support for **one year** (at minimum, perhaps longer). The version you buy is yours to keep and will work forever. Costs will only be incurred for [updates](#), if you want them.

### How to use your unlock key

After you complete the order process, the registration key will be sent to you via email. Once you receive your key by email, click on **I have an unlock key** and start the external license manager in full administrator mode. If you see any UAC elevation prompt click **Yes** to run xplorer<sup>2</sup> elevated.

 As a matter of courtesy to new users xplorer<sup>2</sup> does not ask ('nag') for a registration key during the first day you start using it. If you have a key and you want to use it immediately then start xplorer<sup>2</sup> and use **Help > Activate with key** menu command. If you are using the [ribbon](#) this command is in **License management** menu under the leftmost File ribbon tab.

Open the email with the unlock key, and **copy/paste** the long funny looking key (all of it, usually 3-4 lines of text) and paste it in the license manager window, then click on **Activate license** button. If all goes well, you can restart xplorer<sup>2</sup> and enjoy the program in full power. If you use **Help > About** menu command you will see your name and email as the proud owner of a fully licensed xplorer<sup>2</sup>

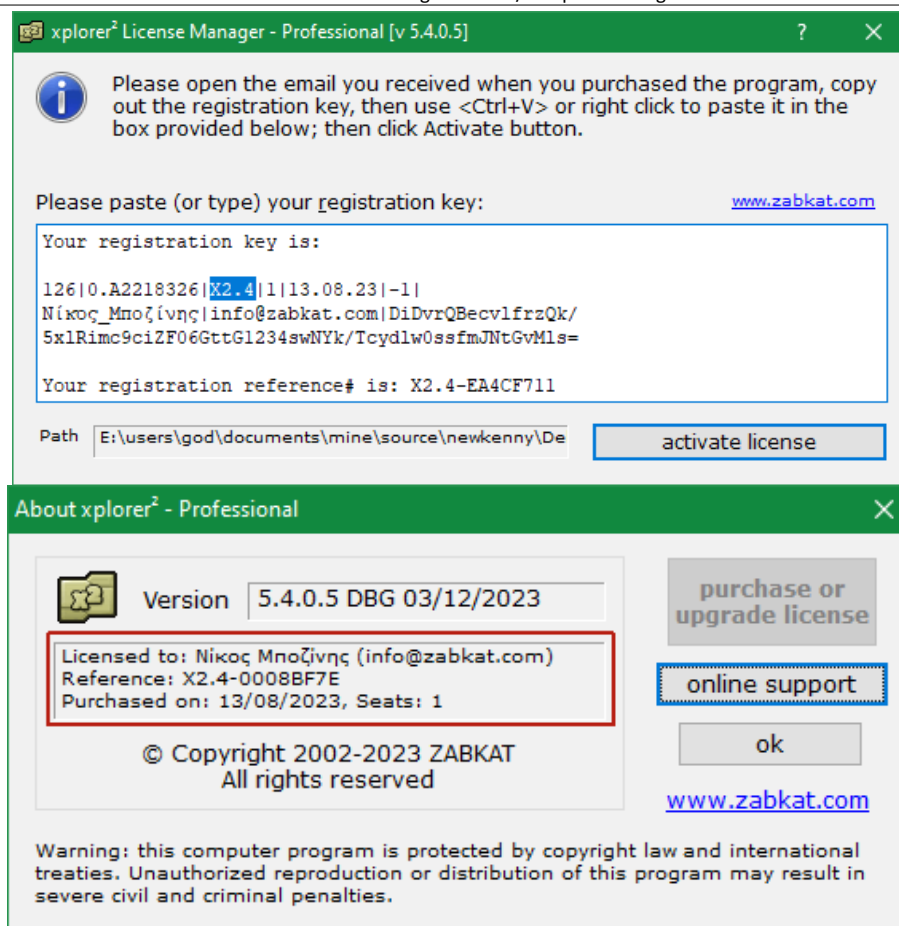



Figure 57. Paste unlock key in license dialog

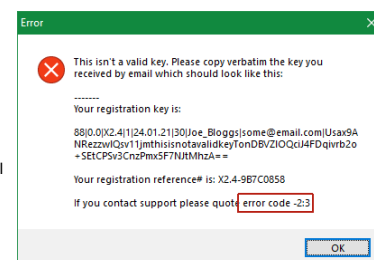
 The professional and ultimate editions have **different unlock keys**. If you bought a license for xplorer<sup>2</sup> professional it can only be used if you install the professional version (and vice versa for the ultimate edition). You can check which xplorer<sup>2</sup> version you are running using **Help > About** menu command — or check the window title bar of the license manager. If you don't have the correct version installed just get the relevant trial version from the website. You can quickly tell which license you bought from the format of the registration **reference number** (see the email receipt). Ultimate version keys are *longer*.

- professional: **X2.4**-12345678
- ultimate: **X2UL.4**-E2282F64

If there's something wrong with your key, xplorer<sup>2</sup> will tell you about it with an error message, which will include a brief explanation of the problem and a numeric **error code** (see the pic to the right). Here are the most frequent error codes and their remedies:

- **Key typo (failed checksum -2:3)**. Did you copy out the **entire** key? It should be 3-4 text lines and look like the one in figure 57 above. Some people leave out the trailing = (equal) signs. Spaces & formatting doesn't matter but you shouldn't modify the key in any way.
- **Key for other program (-3:2)**. If you own other zabkat programs, you will find that all unlock keys look similar, so you may end up using your i-DeClone key to unlock xplorer<sup>2</sup>! Please make sure the key includes the token **|X2.xxx|** (or **|X2UL.xxx|** for ultimate) which indicates a key for xplorer<sup>2</sup>. This error may also indicate you are using a professional key to unlock ultimate edition (or vice versa).
- **Wrong version (expired key -5:5)**. Most likely you are using an older version key to unlock the latest version 5.x. You must either pay for an upgrade or download the [older version](#) you bought.


If you still have problems please watch the [demo video](#) about using the key, and failing even that you can always contact the [support team](#), quoting the exact error code.



## Suspected bugs

xplorer<sup>2</sup> is thoroughly tested and used by many people all around the world, so persistent bugs are few and far between. However being a low-level tool, and given the diversity of operating systems it runs, the multiplicity of user configurations, and buggy [shell extensions](#) people install, crashes and hangs are a sad fact of life, like taxes.

When xplorer<sup>2</sup> crashes for one reason or other, it usually restarts itself automatically. If you see a lot of crashes in similar circumstances, please notify program support using **Help > Crash information** menu command. This tries to retrieve extended information on the latest crash and attaches it to an email meant for xplorer<sup>2</sup> customer support.

 Crash information is a windows minidump file named x2minidump\_xxx\_x64.mdmp, where xxx is your installed version number. Some email clients people use won't allow direct attachments; in that case xplorer<sup>2</sup> adds the line *"Please manually locate and drag-drop the attachment here using the path: \"*. Please locate the MDMP file in %TEMP% folder and attach it to your email report.

Another type of problem is xplorer<sup>2</sup> getting stuck, **not responding**. This occurs naturally sometimes, when xplorer<sup>2</sup> is busy in a heavy computation you've asked for, but if it doesn't come back (after waiting a reasonable amount of time) it could be stuck for good. You can try pressing <ESC> key a few times; failing that, start **another** xplorer<sup>2</sup> instance using its desktop shortcut, and immediately use **Help > Crash information** menu: this should detect that there's a stuck instance and generate an email with information about the problem. Please send it forward to xplorer<sup>2</sup> support.

 This trick cannot help if xplorer<sup>2</sup> is hanging when it **starts**. Many times startup troubles are down to unavailable network drives you were browsing last time, so you can try starting xplorer<sup>2</sup> at fixed safe locations without tabs, using [command line](#) arguments like:  
xplorer2\_64.exe /1 /T c:\

If safe starting folders won't solve the hang, you can try the [procdump](#) tool. A more drastic solution would be to uninstall, erase all your settings, then reinstall. However this will lose all your bookmarks etc


If you think you've discovered a **bug**, please ask yourself a few questions before seeking support:

- Will **rebooting** your PC solve the problem? It is a well known and sad fact that windows isn't the most robust operating system in the world. Many times starting afresh will make the problem go away.
- Try downloading the [latest](#) xplorer<sup>2</sup> version your license will allow. Bugs are fixed all the time, so please keep your version current.
- Is this a **known** issue? Try searching the documentation (including the plain text files, xplorer<sup>2</sup> **Mark > Containing text** command will be helpful) for a relevant keyword. Your bug may be a "feature" after all!
- Can you **reproduce** the bug? If you can't, neither would the development team so they couldn't help you. Some crashes are just freak 1-off ignorable events.
- Think of the **context** when the problem occurs. Is it only for some specific files or folders? What is their common characteristic? Please include as much information as you can when reporting a bug.

Thank you and sorry for any inconvenience caused!

Troubleshooting common problems

Table 9. Problems and solutions

Problem	Solution
I can't use my unlock key	Please copy/paste the entire key from the registration email. For more information see this step by step <a href="#">guide</a>
My unlock key stopped working	Old keys bought for previous xplorer <sup>2</sup> versions will not work for the latest version. You must either get a <a href="#">replacement key</a> (either free or paid upgrade) — or continue using the older version you bought originally.
xplorer <sup>2</sup> is slow to start	Did you leave xplorer <sup>2</sup> browsing network locations that are no longer connected? Try using /N command line <a href="#">switch</a> or start on safe <a href="#">fixed locations</a> .
General browsing is very slow	If browsing folders is painfully slow, you may be using too many of the advanced features. Prime suspect is sorting folder items by a slow property or <a href="#">color coding</a> on slow rules. The more you ask from xplorer <sup>2</sup> the slower things will become! <a href="#">More info...</a>
Safely eject USB drive	Don't just yank USB cables, always right click on the external drive icons (e.g. on the drivebar) and pick <b>Eject</b> or <b>Safely remove</b> from the context menu, and wait for the notification that it is " <i>safe to remove hardware</i> ". xplorer <sup>2</sup> should unlock any resources if you e.g. are browsing a folder on the device. For external hard drives also use the eject button  on windows taskbar.
Can't delete locked files	If you get a <b>File in Use</b> error when deleting files, see if you have the file open in its application. You must first close the file then delete it. If you are previewing the file in xplorer <sup>2</sup> , usually there shouldn't be a problem because xplorer <sup>2</sup> turns off the preview before deleting. However some native preview handlers are buggy and may lock files. In that case turn off the previewer with <CTRL+Q> then restart xplorer <sup>2</sup> to delete the file.
Can't copy files in system folders	If you try copying files into protected folders like c:\Program files you will get an <b>Access Denied</b> error. The workaround is to copy files with drag-drop instead of robust copy. See here for more information on <a href="#">UAC</a>
Audio or video preview doesn't work	xplorer <sup>2</sup> uses an embedded version of Windows Media Player (WMP) for playback. If your videos show without picture or sound, or xplorer <sup>2</sup> crashes on media preview, try downloading a recent media codec pack like <a href="#">K-Lite</a>
I cannot see thumbnails	xplorer <sup>2</sup> shares the fast windows thumbnail cache for quick image previews. Sometimes the cache gets full or corrupted and must be rebuilt. For this and other recovery options <a href="#">click here</a> . An extreme solution is "Don't use windows thumbnail cache" <a href="#">advanced option</a> but then extraction will be slow.


Further reading  
All frequently asked questions ([FAQ](#))

## Appendix

### Keyboard shortcuts

Keyboard is faster than mouse, that's why xplorer<sup>2</sup> is keyboard happy. All menu commands have keyboard shortcuts, and those that don't can be [assigned](#) keys of your choice. This section lists some keyboard shortcuts that help navigating in large folders, and also for tabbing around xplorer<sup>2</sup> window panes.


The <ALT> key modifier is used a lot in any command that is meant for browsing. So whereas pressing <ENTER> on a selected subfolder name will browse into it, <ALT+ENTER> will browse the folder in the *inactive* pane — the active will stay put. <ALT> works the same way with mouse-initiated folder browsing, e.g. from the tree.

 Most of these commands aren't on any menu bar; the command ID column can be used to find them in the **Miscellaneous** category of keyboard customization dialog. Some don't have xplorer<sup>2</sup> command numbers, meaning they are internal and cannot be changed

**Table 10.** Keyboard shortcuts for navigating contents

Key	Description	ID
Arrows	Move selection to the nearest item in the direction of the arrow key	
Home	Move to the first item in a folder	
End	Move to the last item in a folder	
Page Up	Move up one page full of items	
Page Down	Move down one page full of items	
Alt+Up	Move to the previous selected item without harming the selection	32839
Alt+Down	Move to the next selected item without harming the selection	32838
Ctrl+Alt+Up	Move to the first item in the previous <a href="#">group</a>	33110
Ctrl+Alt+Down	Move to the first item in the next group	33109
Ctrl+Shift+Left	Collapse all groups	33200
Ctrl+Shift+Down	Move active list item down (temporary)	33103
Ctrl+Shift+Up	Move active list item up (temporary)	33104
Ctrl+E	Toggle between two equal and one large view pane	33032
Tab	Switch to inactive folder pane	32857
Ctrl+Tab	Switch to tree pane	32858
Ctrl+Alt+Tab	Switch to mini-scrap pane	32861
Shift+Tab	Switch to address bar	32859
	Switch to the filter box (addressbar)	33250
Ctrl+Shift+Tab	Switch to quick preview pane	32860
Ctrl+Shift+F6	activate next window pane	33525
	activate previous window pane	33526
Ctrl+Alt+Left	Switch to previous view tab	33039
Ctrl+Alt+Right	Switch to next view tab	33038
Ctrl+Alt+Z	Toggle quick filter or quick search <a href="#">mode</a>	33542
(customize)	Toggle between the Draft and Native tabs of <a href="#">QuickViewer</a>	33050
Ctrl+Y	Repeat last menu command	57640
Ctrl+Alt+F10	Repeat the last shell context menu command (eg. TSVN submenu)	33125


These keyboard shortcuts are the default assignments, assuming you haven't changed them already!

 When it comes to jumping into various panes, there are both direct jump commands e.g. <SHIFT+TAB> jumps to the addressbar, and **cycling** commands (33525 activate next pane) that jump to neighboring panes in a ring. If you cannot remember the exact TAB shortcut that takes you some place, use cycle a few times to get there!  
Press <ESC> key from any peripheral pane to return back to active folder pane.

## Stock properties

The modern windows file [property system](#) is rich and mature, but things weren't always this way. xplorer<sup>2</sup> stock properties provide a consistent set of basic and advanced properties, that work in both old and recent Windows OS versions. Moreover they work both in regular filesystem and [special](#) folders like compressed zipfolders. So whenever you can, use a stock property instead of the system equivalent, for a consistent browsing and searching experience.

Stock properties appear first in column selection [dialog](#) and are recognized by the [S] symbol after their name. The first few are duplicates of basic file properties like name, size and date, but you will find lots of interesting extended properties that are unique to xplorer<sup>2</sup> like digital signature information for executables.

 If you need some other file property not covered by stock and windows columns, have a look at [WDX plugins](#) or legacy [column handlers](#) — all of which appear with [X] symbol in the column dialog. If you are a programmer it is relatively simple to create your own WDX plugin (much easier compared to a shell column handler) for that special file property you cannot find elsewhere!

**Table 11.** Stock file properties

Name	Description
Name	Display name (may not include extension depending on your folder options)
Full name	Actual file name
Extension	Filename extension (without the dot)
8.3 Name	DOS 8.3 <a href="#">short name</a> , guaranteed without spaces
Size	File size
Size on disk	Actual size occupied on disk (including compressed/sparse files and sector allocation)
Type	File type, e.g. "Text document"
Attributes	File Attributes
Modified	Date last modified
Created	Creation date
Accessed	Last accessed date
Path	Full path of container folder (not including the file name), useful for search results
Medium	Data medium information (hard disk label)
Contents	Number of direct subitems in a folder, can be used to detect <a href="#">empty</a> folders
Characters	Character length of full pathname, for tracking extra <a href="#">long names</a>
Infotip	Extended file type specific information (what you see in the popup tip but in a searchable column)
Links	Number of <a href="#">hard links</a> (=1 if there are no associated links)
Shortcut to	Path to (real) target object associated with LNK shortcut or junction
Checksum	Simple checksum of contents (xplorer <sup>2</sup> own flavor)
MD5 hash	MD5 digest for data integrity (slow, examines all the file up to a <a href="#">limit</a> )
Comment	Extra <a href="#">user information</a> , can be used for all file types
Keywords	Document related keywords
Streams	Alternate data streams ( <a href="#">ADS</a> ) counter (=1 if just the contents stream exists)
Stream names	Named streams attached to file (ADS) if any
Version	Executable program or DLL version
Company	Company owning the rights of binary code
Description	Module extended information
Signed by	Authenticode digital signature, for secure installers
Width	Picture width in pixels (EXIF) — applies to all image types
Height	Picture height (EXIF)
Picture date	Date picture was taken (EXIF) — only for JPG camera photos
Blur	Photo <a href="#">blur</a> (bad quality probability if greater than 100)
Group ID	Logical group identifier (for <a href="#">duplicates</a> )
Color tag	Index in color tag array; numeric equivalent of <a href="#">color border</a> (0=untagged, then 1,2,...)
Programmable	Combination of file properties ( <a href="#">ultimate</a> only)

## editor<sup>2</sup>

The internal text viewer and editor is a lightweight notepad replacement, very capable for simple editing — the present xplorer<sup>2</sup> manual was composed in editor<sup>2</sup> ! It is a separate program distributed together with xplorer<sup>2</sup>, and handles by default text and RTF documents with **File > View** menu command in xplorer<sup>2</sup>.

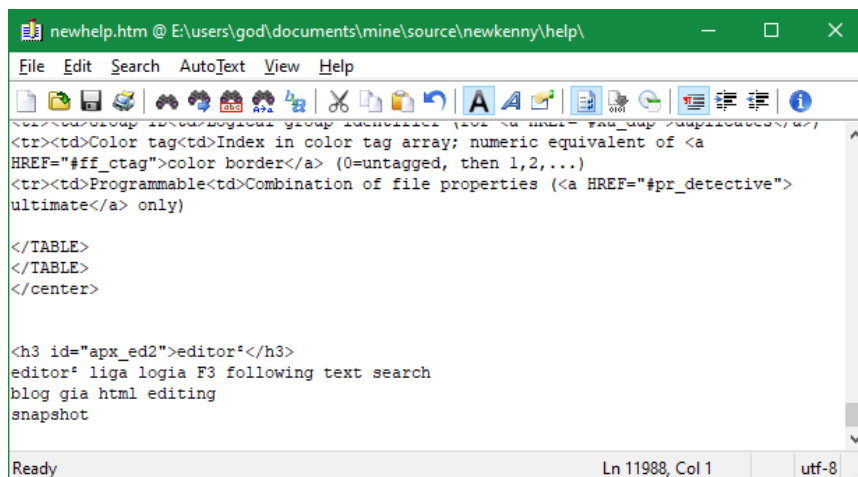


Figure 58. editor<sup>2</sup> main window

It is a simple program with a menu, toolbar, statusbar and the main editing area. It is quick but surprisingly powerful. It understands all popular text **encodings** (plain, UTF8, unicode, custom codepages) and adjusts file load and saving accordingly. It is customizable, setting fonts, colors, tab sizes etc with **View > Options** menu. Here are some handy features for simple HTML and source code editing:

- **Indentation.** Automatically align code blocks with **View > Auto indent** menu, and move selected blocks in and out
- **Match brackets.** Select an opening (or closing) bracket or brace, and find its partner with **Search > Match brace** menu
- **HTML tags.** Similar to brace matching, you can find the matching closing (or opening) tag if you click on the opening one and use **Search > Match tag** menu. Thus you can find the matching pair of <DIV> </DIV> or any other HTML tag. Hold down <SHIFT> to select the entire block.
- **2 bookmarks.** Use **Search > Set bookmark** menu to set a bookmark at the current caret location, then press <F8> to quickly return there. You can set 2 such bookmarks to jump around (new bookmarks overwrite the oldest one). Press <SHIFT+F8> to extend the selection to the nearest bookmark.
- **Autotexts.** With a single click or key you insert commonly used text like HTML tags. You define your own with **Autotext** menu
- **Macros.** The same UI [scripting](#) language is available in editor<sup>2</sup> to do bulk manipulations of similar lines. Not for the faint hearted

editor<sup>2</sup> is closely integrated with xplorer<sup>2</sup> text searches. Whatever you search in xplorer<sup>2</sup> is communicated to editor<sup>2</sup> (and vice versa), so if you recently searched for [text in files](#) then open one of the hits in editor<sup>2</sup> pressing <F3>, another <F3> (**Search > Find next** menu) will locate the keyword you searched in xplorer<sup>2</sup>!



## All menu commands

Here you will find all main window menu commands with brief explanations of their use and [cross linking](#) with sections in the manual where they are explained in more detail.

**Table 12.** Main window menu commands

Command	Description
<b>File</b>	
Browse	Browse folder or launch item
Clone+browse	Browse the focused item in a new window
Enter archive	Try to browse archive item as folder (requires WCX plugin)
New tab	Browse the focused item in a <a href="#">new tab</a>
Column mode	Browse folder trees using side by side ( <a href="#">Miller</a> ) panels
Browse flat	Open a scrap container for selection (folder or cida) and add all contents <a href="#">recursively</a> (<CTRL> to reuse an existing window)
Send to scrap	Place selection in the nearest scrap container window
View	View file contents as <a href="#">text</a>
Edit	Edit file contents as text
Shell edit	Execute the edit shell verb on the selection (type specific)
Properties	Displays the shell <a href="#">properties</a> of the selected item(s)
Rename	Rename the selected item in place
Change type	Rename an item including its DOS extension
Mass rename	Change selected filenames <a href="#">in bulk</a> using name templates
Delete	Send selected items to the recycle bin (<SHIFT> for permanent deletion)
Close	Close this window
Exit	Quit the application (all open windows)
<b>Go to</b>	
Up a level	Browse one level up from the current folder
Back	Return to the folder browsed before
Forward	Advance to the next folder in list
Find target	Find target item pointed to by this shortcut file (resolve link)
Select drive	Goto a local disk
Recent folder	Select among folders recently browsed in this pane
Drop target	Folder where you last drag-dropped or pasted something (helps undo <a href="#">accidental drops</a> )
Special folders ►	
My documents	Contains personal files
My pictures	Digital photographs
My music	Audio files (MP3 etc)
Downloads	Files recently downloaded
Recent items	Documents and folders accessed (opened or saved) recently
Recycle bin	Deleted documents
Desktop	Namespace root
Root	Root drive of current folder, e.g. C:\
My computer	Local and mapped drives (This PC)
Network	Computers near this
Control panel	Computer settings
Printers	Available printers
Start menu	Folder representing the Start button
Startup	Programs that start every time you logon
CD burning	Files waiting to be written to CD (XP or later)
Send to	Targets appearing in the Send To shell context menu
Compatibility files	Redirected folder for legacy applications (Vista or later)
Quick bookmark	Go to the folder bookmarked earlier
Set quick-mark	Set <a href="#">quick-bookmark</a> on the current folder
Frequent ►	
(...)	Browse this frequently used folder
Mirror browsing	If checked, the panes <a href="#">follow each other</a> as you traverse the folder hierarchy
Mirror scrolling	Automatically scroll the inactive pane to reveal items matching the active focus
Same folder	Force inactive pane to the same folder as the active one (<ALT> for the inverse)
Swap panes	Mutually exchange the contents of the two panes
<b>Bookmarks</b>	
Dual ►	
Add current	Add left/right folders in the <a href="#">dual bookmark</a> list
Organize	Rename, delete or reorder dual bookmarks & assign shortcut keys
(...)	Restore these two folders
More	Select folders from the complete dual bookmarks list
Add current	Add current folder in the bookmark list
Organize	Rename, delete or reorder bookmarks & assign shortcut keys
(...)	Go to this folder
More	Select a folder from the complete bookmarks list
Internet Favorites	Go to the folder holding internet explorer's favorites
Explorer Favorites	Favorites (links) folder as included in windows explorer

Table 12b-->

Table 12b. Main window menu commands (cont.)

Command	Description
<b>Mark</b>	
Select All	Select all items
Unselect all	Unselect all items
All folders	Mark all items that are assumed folders
Sticky selection	Toggle sticky item selection mode (like <a href="#">checkboxes</a> , unaffected by clicks or keys)
Select group	Select items that match a <a href="#">wildcard</a> filter
Unselect group	Unselect items that match a filter
Invert selection	Select unselected items and vice-versa
Select range	Select a number of items below the active one
Total size	Mark items whose combined size approximates a given total byte size
Synchronize panes	Select items that differ among the panes, using date information
Sync wizard	<a href="#">Synchronize</a> panes with tailor-made options
Check build	Mark items according to file <a href="#">transformation</a> rules (e.g. compilation)
Containing text	Select files that contain some specified <a href="#">text string</a>
Matching a rule	Select items whose properties match a <a href="#">complex rule</a>
Quick search	Select items with text in either filenames, columns or file content
<a href="#">Selection</a> ►	
Store	Remember the selected items for later use
Select	Select all items stored earlier in the selection clipboard
Unselect	Unselect all items stored earlier in the selection clipboard
Combine	Combine the existing and saved selections in a boolean AND fashion
Show	Bring selected items into view
<b>Edit</b>	
Cut	Cut the selection and put it on the Clipboard
Copy	Copy the selection and put it on the Clipboard
Copy to	Transfer selection <a href="#">robustly</a> to another folder (<ALT> for last target)
Move to	Move selection to another folder (<ALT> for last target)
Queue status	Examine and organize the <a href="#">queued</a> transfers
Paste	Insert Clipboard contents
Paste Link	Create shortcuts to the items held in the clipboard
<a href="#">Paste special</a> ►	
Folder structure	Paste all and only subfolders without any files they contain
Structured scrap clips	Paste items (sourced from a scrap container) recreating the original subfolder structure
Multi paste	Insert clipboard contents on all selected folders
Hard link	Create hard links for the items held in the clipboard (NTFS only)
Folder junction ►	
Paste	Create a junction here pointing to the folder in clipboard
Delete	Remove folder junction without harming the target folder
Symbolic link	Create symbolic links for the items held in the clipboard (Vista or later)
Duplicate	Create a copy of the selected item(s) in the same folder
Copy path names	Copy the <a href="#">full path</a> names of the selected items to the clipboard (<SHIFT> for short 8.3 paths)
Copy filenames	Copy selected filenames, without the folder path
Copy preview	Copy item's preview image as bitmap
Copy columns	Copy selected items' column text as it appears in the active pane (for printing etc), <ALT> for highlighted column only
Extract text	Extract unformatted text from selected document to a new file
Export CSV	Export view contents in <a href="#">CSV</a> text format for Excel

Table 12c--&gt;

Table 12c. Main window menu commands (cont.)

Command	Description
<b>View</b>	
Dual pane	Toggle single/dual folder pane mode
Tile horizontally	Align dual panes left/right (untick to align up/down)
Ribbon	Show the <a href="#">ribbon</a> UI or a traditional menu bar
Quick viewer	Show or hide the quick <a href="#">previewer</a> pane
Lock viewer	If checked the previewer will stop following the active item, freezing on the last item
Peek preview	<a href="#">Popup</a> preview of the focused item (click both mouse buttons)
Toolbars ►	
All toolbars	Show or hide the toolbars' container (rebar)
Status Bar	Show or hide the <a href="#">status bar</a>
Info bars	Show or hide detailed <a href="#">information</a> bars for the active item (file)
Lock dock sites	Fix dock panes to prevent their rearrangement (toggle)
Details pane	Show or hide the item <a href="#">details pane</a>
<a href="#">Pane style</a> ►	
Large icons	Displays items using large icons
Small icons	Displays items using small icons
List	Displays items in a list
Details	Displays detailed information for each item
Thumbnails	Displays preview icons for items
Mini thumbs	Displays smaller previews for items
Detailed thumbs	Displays item details and a thumbnail preview
Select columns	Organize <a href="#">columns</a> for detailed view mode
<a href="#">Arrange by</a> ►	
Name	Sort alphabetically by name
Size	Sort by size, from smallest to largest
Date	Sort by modification date, from most recent to oldest
Type	Sort by filename extension
Other	Sort by a non-standard column (through header row click)
Unsorted	Do not sort items; display them in the order read from disk
Ascending	Toggles sorting from smaller to largest
Manually	Arrange icons using the mouse ( <a href="#">desktop-like</a> )
Randomly	Item positions picked at random
Show in groups	<a href="#">Group items</a> in the window by the sorted property
Custom groups	Group items using a complex filter for each category
Autosize columns	Adjust column widths to fit all items (<SHIFT> to use headers)
Refresh	Refresh pane contents (usually done automatically when changes are detected)
Hold autorefresh	Temporarily suspend the active pane's autorefresh sensor, useful for slow or fast-changing folders
Raw contents	Browse a virtual folder exactly as it is stored on disk (bypass explorer); try it on the system <code>Fonts</code> folder to see what it does.
Show Tree	Show or hide the <a href="#">folder tree</a> pane
Locate in tree	Locate the browsed folder within the tree hierarchy
Mini scrap	Show or hide the temporary item hold area ( <a href="#">drop box</a> )
Visual filter ►	
Wildcard	Constrain item visibility according to supplied wildcard
Rule based	Show only those items whose properties match a <a href="#">complex rule</a>
Auto-filter	Show only one file type from <a href="#">those present</a> in this folder
On/Off	Toggle the wildcard- or rule-based filter on/off
Hide folders	Hide all folders from the active pane (toggle)
Selected only	Hide all items that aren't selected
Same filter	Force the same visual filter to the inactive panel
Show all	Cancel all visual filtering modes, showing all items

Table 12d--&gt;

**Table 12d.** Main window menu commands (cont.)

Command	Description
<b>Actions</b>	
New file	Create a new (empty) file
New Folder	Create a new folder
Set comment	Set, edit or clear item's <a href="#">comment</a> , if possible
ADS ►	
View streams	Show any <a href="#">alternate data streams</a> contained in the selection
Split streams	Extract secondary streams from the active item
Bundle to go	Pack all (selected) file information for transport, including ADS comments etc
Unpack bundle	Restore file information from an existing bundle
Change attributes	Change dates and <a href="#">DOS attributes</a> of the selected items
Sync-Touch	Synchronize modification dates of matching items (from active to inactive panel)
Change dates	Various operations on <a href="#">file dates</a> (shift, touch etc)
File properties	Change text file properties (Author etc) in bulk
Split file	Split the selected file into <a href="#">smaller chunks</a>
Merge files	Combine selected items into a single file (in the order shown in pane)
Shred	Total <a href="#">annihilation</a> of the selected items
Save settings now	Save all current settings immediately
Export settings	<a href="#">Export</a> program settings in a .REG file that can be reused in another PC
Import settings	Restart xplorer2 to import settings saved earlier
<a href="#">Folder settings</a> ►	
Save	Save settings for this folder, including pane style, columns and sort order
Clear	Clear all stored settings for this folder
Suspend	Temporarily ignore all custom folder settings for this view pane (toggle)
Default	Show all folders like this one; untick for variable folder view settings
<b>Tools</b>	
Run command	<a href="#">Start</a> any windows (GUI) program
DOS command	Run a command that requires the DOS interpreter (dir, ren, etc)
Run History	Select a command from the history list
Repeat command	Quick-repeat of the last command executed
Command script	Creates a script ( <a href="#">batch file</a> ), applying a command template on each selected item
Find files	<a href="#">Look for</a> files and folders that match a filter
Fast search	Quick search using windows <a href="#">indexing server</a> , under current folder
Check duplicates	Examine items under current folder for possible <a href="#">duplicates</a>
Compare subfolders	Compare pane contents considering all <a href="#">subfolders</a>
Map network drive	Assign a drive letter to a network resource
Backup copy	Copy newer and <a href="#">changed files</a> only, to inactive pane (with subfolders)
Free space	Check the free space in the disk(s) being browsed; refresh views if necessary
Subfolder size	Calculate total size occupied by folders and their <a href="#">subfolders</a>
Folder statistics	Display <a href="#">statistics</a> for this folder and its subfolders
Options	Configure <a href="#">options</a> for the program & this window
Advanced options	Open an external program to tweak <a href="#">advanced</a> program settings
<b>Window</b>	
Scrap container	Open a <a href="#">scrap container</a> for temporarily holding items from various folders
List	List all windows controlled by this application; optionally switch among them
Administrator	Start an elevated xplorer <sup>2</sup> window to bypass UAC (access permissions)
Close tab	Close the active folder tab
Close all tabs	Close all folder tabs except for the active one
Reopen tab	Reopen the most recently closed tab
Find tab	Search and locate a tab by its path
Activate tab ►	
(...)	Switch to this tab
Clone	Open another explorer window like this one (runs in its own thread)
Save layout	Store the present window <a href="#">layout</a> and history settings
(...)	Browse the active item in a new window using this saved layout
Organize	Assign shortcut keys for or delete window layouts
More layouts	Clone a window from another saved layout (not showing in the above list)

Table 12e--&gt;

Table 12e. Main window menu commands (cont.)

Command	Description
<b>Customize</b>	
Toolbars ►	
Add new	Add an extra toolbar
Organize	Rename or delete extra toolbars
(...)	Add or remove buttons from <a href="#">this toolbar</a> (right-click on it for text options)
Column sets ►	
Add current	Add the current <a href="#">column set</a> to the list
Organize	Rename, delete or reorder column sets & assign shortcut keys
(...)	Restore this column set
More	Pick a column set from the complete list
User commands ►	
Add new	Add a <a href="#">command template</a> to the user menu
Add current	Add the currently selected executable program as a user command
Organize	Rename, delete or reorder user commands & assign shortcut keys
(...)	Execute this command
More	Pick a command from the complete list
Folder groups ►	
Add current	Add the current folder group ( <a href="#">tab set</a> ) to the list
Organize	Rename, delete or reorder folder groups & assign shortcut keys
(...)	Restore this folder group
More	Pick a folder group from the complete list
Startup	Restore original folder tabs present at program startup, to recover previously unavailable network folders
Custom groups ►	
Add current	Save the current view <a href="#">categories</a> in the list
Organize	Rename, delete or reorder custom groups & assign shortcut keys
(...)	Restore this custom group
More	Pick a custom group from the complete list
Macros ►	
Record macro	Record new <a href="#">macro</a> command (<CTRL> to repeat last macro)
Edit macro	Start the macro editor
Organize	Rename, delete or reorder <a href="#">macros</a> & assign shortcut keys
(...)	Run this macro
More	Pick a macro from the complete list
Keyboard	Change <a href="#">keyboard</a> shortcuts to menu commands
Keyboard use	View assigned and available keyboard shortcuts
Color coding	Customize item <a href="#">appearance</a> according to file type
<b>Help</b>	
Quick start	Introductory guide to xplorer <sup>2</sup> (shows <a href="#">this</a> document)
Tip of the day	Tips & <a href="#">tricks</a> for maximizing your exploring experience!
How do I?	Show or hide the on-board tutorial pane with demo videos (requires internet connection)
Command finder	<a href="#">Locate</a> a menu command by matching a keyword
Activate with key	Use the registration key you purchased to <a href="#">unlock</a> the program
Register program	Upgrade and/or obtain a license to use the pro version without restrictions
Upgrade to ultimate	Power up your xplorer <sup>2</sup> experience getting the <a href="#">ultimate</a> edition
Donate	Support the development of this program by making a voluntary donation (free lite version only)
Check for updates	Goto the website for the <a href="#">latest</a> program build
Online support	Get connected to the online customer support centre
Translation	Toggle english translation on/off (assuming you added a language during <a href="#">installation</a> )
Last error	Show the last error message again
Crash information	Locate the <a href="#">crash</a> minidump file; email as attachment to customer support
About xplorer <sup>2</sup>	Display program information, version number and copyright
Licence agreement	Read the terms & conditions associated with using the program
What's new?	See a list of recent changes and added features
Tell a friend	Recommend xplorer <sup>2</sup> to a friend — thanks for spreading the word!

Hint : [Scrap](#) container windows have almost the same menu commands, except for those special in the table below

Table 13-->

Hint: [Scrap](#) container windows have almost the same menu commands, except for those special in the table below

Table 13. Scrap window special menu commands

Command	Description
<b>File</b>	
Browse	Load the selected items into a regular folder pane (open <a href="#">container</a> )
Remove	Remove selected items from this pane (real files not affected)
Clear	Empty all pane contents
<b>Edit</b>	
Sync-o-paste	<a href="#">Enforce</a> deep synchronization results, copying selected items to their respective target folders
<b>View</b>	
Arrange by ► Path	Sort by the parent folder's path
<b>Actions</b>	
Load contents	Open CIDA <a href="#">content file</a> written earlier
Write contents	Save active pane's contents in a file for later retrieval
Recent File ► (...)	Open this CIDA document
Import clipboard	Insert items from <a href="#">clipboard names</a> (one full path per line)
<b>Tools</b>	
Search status	View a <a href="#">log</a> of an ongoing or past find files command
Save search	<a href="#">Save</a> last search parameters to a file, to repeat later
Recent search ► (...)	Repeat this search
Reveal unique	Reveal items hidden as a result of a duplicates check (like <b>View &gt; Show all</b> menu)
Find similar photos	Group JPG pictures according to their <a href="#">similarity</a>
Fuzzy groups	Group items according to a <a href="#">similar name</a> , or other text attribute
Extract groups	<a href="#">Create folders</a> out of group or date information

◀ [Appendix](#) | [Contents](#)