# 2xExplorer R7x

*A swiss army knife for your everyday file management needs*

Windows Explorer is a decent entry-level file manager, capable of allowing even one's grandmother to become an instant file manager-ess. The point and click, drag & drop paradigm is very intuitive and gives access to immediate no-frills basic file management. However, for the more dedicated user who owns his/her computer for more than a month or two, and whose interactions with the machine extend beyond surfing the net, the fundamental cracks in the said explorer become apparent.

Have you ever wished the standard Windows Explorer was a little bit more efficient for the day-to-day file management tasks? Are you longing for the functionality of older file managers but can't find one that is 32-bit compatible? Do you find some applications in the explorer replacement market a bit too taxing to setup and use? Then look no further, ***2xExplorer*** is the answer and twice as nice at that. It combines the user-friendliness of the standard Explorer with the increased efficiency and advanced features of good-old **dual-pane** file managers like *Norton Commander*. It is a package that will appeal to the power user, yet its similarity with Explorer will ensure that less experienced users will not be driven away either.

2xExplorer is a fusion between the efficient dual-pane file management format of yesteryear and the intuitive tree/content pane format of modern operating systems. It aims to deliver the best of both worlds, without pledging allegiance to either of the two prototypes. You may regard it as an explorer on steroids, with an additional pane for browsing folder contents side to side. Apart for this extra pane, 2xExplorer fills in many glaring *omissions* in the windows Explorer:

- **Filters** (wildcard based) for limiting the files displayed in a contents pane, and for controlling the selection of items.
- Keyboard **shortcuts** for nearly all commands, an indispensable feature for efficient everyday use.
- Briefcase-beating lightweight folder **synchronisation** engine.
- Notepad-thrashing internal simple yet powerful text **editor** and RTF/Hex viewer.
- Improved Find Files tool offering more *control* over the search criteria.
- **Printing** of folder contents and other statistics.
- Automatic Batch file (`*.bat`) generation that can be used for delicate management operations, as the **mass renaming** of files.
- Complete file date information (modified *plus* created and last accessed) is both shown and used for various operations throughout the program.
- Proper sorting for file type using the filename extension alphanumerically; no more `".INI"` files before `".BMP"` ones like in the confused explorer.

2xExplorer offers a host of powerful commands found in heavyweight file managers like Norton Commander, yet it is not poised to reproduce faithfully the original item. The author makes no apology for departing from the traditional looks and modus operandi that some NC purists would regard compulsory for any file manager braving a comparison with the founder of the dual pane scene. For example, there seems to be little justification for two sets of toolbars (i.e. drive buttons, status bars, etc.) - one for each pane. Despite this external dissimilarity though, you'll soon realise that 2xExplorer is every bit as efficient, useable and addictive as NC. In fact it surpasses the master in providing all mod cons of an up to date 32-bit shell namespace browser:

- Browsing and file management in *non-filesystem* folders like Control Panel, Desktop, etc.
- Full **drag/drop** (including right-dragging) and shell **context menu** support.
- Multifunctional address bar for specifying folder names *and* visual filters, plus running (executing) DOS or windows programs.
- **Bookmarks** and internet-style back/next folder navigation.
- Separable tree view; one can have *both* the tree view and the two contents panes visible at the same time.
- **Thumbnail** previewer for many popular graphics formats (jpeg, gif, bmp, etc.), HTML and Office documents.
- Handy tools like the Folder Subtree Statistics and the *unique* Mirror Browsing tool.
- **Autocompletion** for paths and other frequently typed information.

All this functionality does not come at the expense of user friendliness. In most respects this is just another Windows Explorer, and whoever is familiar with that, will feel immediately at home using 2xExplorer. The look is not far off the standard Explorer either; one will even see the same "animations" in copying/deleting operations. Descriptions of commands are readily available in the form of explanatory text on the status bar and tooltips popping out of toolbar buttons. Dialog windows have embedded brief instructions of use - with complimentary icons too. 2xExplorer is capable of making instant power users out most people, regardless of their level of experience.

All the features of 2xExplorer are supported for *all* 32-bit windows platforms, from win95 all the way up to the latest win2000 (i.e. 95/98/NT4/2000/Me). The code is fairly **stable** and you shouldn't experience many mysterious program crashes and blue screens of death. Keep in mind that 2xExplorer works at a very low level, hence crashes cannot be ruled out altogether, especially for systems that are heavily tweaked.

Last but not least, 2xExplorer offers all this increased functionality whilst retaining a "small" executable file size. Half-megabyte programs would be regarded monstrous 10 years ago, but with all these COM and user interface features nowadays 500 kB is at the low end of the market. Still don't let that fool you, there are *heaps* of functionality efficiently packed in this lightweight outfit. The file size was kept in check by sensible program flow design, taking advantage of many C++ language features for the maximum level of code reuse possible. This same feature explains in part the **speed** of the program, too.

## Dual Pane Layout

Anybody that has ever worked with a dual pane file manager will surely attest to the one thousand plus one different uses that the extra contents pane can be put into, for increased efficiency. In 2xExplorer the contents of *two* folders are visible at any given time. Each of the two panes behaves similarly to the right hand side pane of the standard Explorer, and can be configured completely separately in all property departments (e.g. view style, sorting method etc.).

However, the dual pane layout is much more than the sum of its parts. It's not the same as merely placing two explorer windows side by side, since the panes can *communicate* with each other. Here are some manifestations of this functionality:

- *One stop copy/move operations*. Perhaps you find the separate "copying" followed by "pasting" too big a fuss to endure; perhaps you want to see for yourself and verify where your files end up to. 2xExplorer covers all these issues. The presence of two panes provides straightforward target and source folders for these operations. The active pane (be it the left or right one) is always the source folder. The target will usually be the "other" (opposite) pane; however, an alternative folder can be specified if necessary. Hence a copy or move operation can be carried out with a single keystroke.
- *Folder synchronisation*. The panes can compare their respective contents and mark all files that are missing or different, so as to bring the two folders in perfect harmonisation. For more information refer to the synchronisation section below.
- *Mirror browsing*. Assisting the synchronisation/comparison of whole directory subtrees, this is a **world-first** feature delivered by 2xExplorer. When turned on, the inactive pane will mirror the hierarchy navigation applied to the active pane. Let us consider an example: initially, "`C:\Documents\Project`" is shown in the active pane (A) and "`A:\`" in the inactive pane (I). With mirror browsing, moving (A) down into "`C:\Documents\Project\Resource`" will result in the (I)-pane *automatically* switching to "`A:\Resource`", should such a folder exist. If you hit `<Backspace>` to go one level up then (I) will go one level up too.
- *Synchronised scrolling*. Activating this option results in tracking the focus cursor ("selected" item) movement in the active pane and trying to match it in the inactive pane. The match aims to locate items with the same name that can be thus individually compared head-to-head. Turning both panes into the Details view mode will maximise the usability of this feature, since 2xExplorer will even try to align both items horizontally, if possible.

The team work between the panes goes even further with commands that load the same folder into the inactive pane, using the `<Alt>` modifier (see *Efficient Folder Navigation* section below) to force-feed its address, and so on. Once you've sampled the advantages of the dual pane format, there's no going back to half measures any more.

## File Management

2xExplorer is a beast of burden ploughing through all your day to day file management needs. You can do all the usual operations like copying, moving, deleting and renaming, with either the mouse or the keyboard, just like the standard Explorer. No need to reconfigure your mind and/or take special training courses there. However, the similarity stops here. 2xExplorer is capable of taking on large folder hierarchies and efficiently managing large amounts of files. It will allow you to get swiftly where you want and increase your productivity once you get there.

There are two main concepts that facilitate managing large scale folders: visual filtering and a multitude of item selection mechanisms. 2xExplorer uses **filters** to constrain the number of items displayed in each pane. Traditional wildcard patterns are recruited for this, using the special characters `"*"` to match any number of any characters and `"?"` to match any single character, like in the good-old command line days. Multiple patterns can be easily specified in a comma separated list, e.g. the filter `"*.cpp, *.h"` will result in showing all files having 'cpp' or 'h' extensions and nothing else.

For even greater control, it is possible to **exclude** certain kinds of files, by specifying a leading minus '-' character in some substring. For instance `"*.cpp, -a*"` will select all 'cpp' files *except* those whose names begin with 'a', and `"-*.txt"` would select all files except those with 'txt' extensions. This *comma-separated list* concept is a great efficiency boosting gadget that is used throughout the application, solving all sorts of problems.

All the file management operations require selected items to act upon. 2xExplorer comes in handy in this department too, with its virtual **selection engine**. Items can be picked individually if clicked with the mouse (possibly while holding down the `<Ctrl>` key) or with the `<Ins>` key. Larger selections are possible with `<shift>`-clicking, lasso selection and all the other means supported by the standard explorer. As you'd expect there's more on offer, in the form of selection *criteria*:

- Filters may be used for selecting and unselecting items out of those that comply with the visual filter specified earlier. The comma-separated filter list is applicable here too. Note that selection filters are totally independent of visual ones.
- Using the range of synchronisation commands (see next section) items can be selected if found to differ among the two panes shown.
- Items can also be marked if found to be containing some user-supplied text. Although not a full-blown `grep` (familiar to UNIX users) equivalent just yet, the command accepts a large degree of tweaking with options to control the case sensitivity of the search, for excluding binary files from the scan, etc.

Complementing these selection/unselection methods are various small touches like inverting the selection, storing the selected names for later use, specifying whether subsequent selections will be added on top of the present one, and more. Ingenious use of these small building blocks will tackle any arbitrarily complex selection formula - limited solely by the user's imagination.

2xExplorer is fully aware of shell **shortcuts** (links) to files and folders, and handles them correctly. For example, if you drag-drop some files on a folder shortcut, the files will be properly sent to the real target folder. Also the `<Ctrl+L>` FindTarget command is very useful for quickly locating the target item of any shortcut file.

## Folder Synchronisation

Personally, I have suffered so many devastating blows during my long-term symbiosis with computers, that have developed a nearly-psychotic tendency to backup every piece of important working file. Note that I am not referring to once a month - or even weekly - backups; if some dark cosmic eventuality causes my losing even one hour's worth of work I feel gutted. It is the well known universal law of the sod that things will pack up the very moment that absolute reliability is at a premium. The text editor will delete the thesis you were working on, the developer studio will crash and corrupt your C++ source files, even the operator himself is liable to malfunction after long hours and do the daft thing. These are all too familiar sights that are capable of driving even mother Teresa through the wall, if under pressure.

The answer to these kinds of hiccups is to do frequent "backups" onto a floppy disk, at a rate inversely proportional to your insecurity factor - that's every 10 minutes if your name is Nikos. For that purpose you need a very efficient and fast tool that can automatically detect the files you've modified and quickly copy them to their secure haven. Well, you've guessed it, 2xExplorer does just that with a variety of synchronisation commands.

You don't have to share the author's paranoia to appreciate the synchronisation features. Any person that works in a couple of different locations will need a means of keeping the contents of the two computers in sync. If you want to continue your labours at home after a long day's work, you'll have to transfer your project files via a floppy disk. Given the sluggishness of file transfer to/from floppy media, the capability to detect and copy only modified files is invaluable. 2xExplorer does not suffer from the same limitations as the Windows Briefcase which is badly constrained by binding to two *specific* fixed folders. The synchronisation of any two folders can be examined in a flash using the following palette of commands:

- *Plain synchronisation*. Common as garden grass, this is a very quick way to detect modified files, by looking at the date stamp information of all filenames that exist in both panes (new files that exist only in one pane are straightforward subjects for synchronisation). A later date will result in this file considered more recent and thus marked by 2xExplorer. Note that the files are only marked as *suggested* for copying; it is up to the user to decide what to do with them. Copying would be the sensible option but nobody could stop you from deleting or performing any other sensible action.
- *Contents synchronisation*. Sometimes the date stamp information is inadequate for comparing file versions. For instance, copying files from a windows PC to a UNIX computer messes up the modification dates; FTP transfers may suffer from the same symptom. In such cases the last modification date is not a reliable indicator, and the **contents** synchronisation comes to the rescue. This is a slower method since files have to be physically opened and their contents read in memory to determine their differences or otherwise. Moreover when two files are found to differ, 2xExplorer has no means of telling which one is the most recently reviewed version, therefore only files in the active pane are marked. It is the user's responsibility to discern which version to keep and which to junk. Despite these inevitable shortcomings, content synchronisation can be invaluable when the going gets tough.

2xExplorer offers *lightweight* synchronisation as opposed to full-blown backup services. The emphasis is on efficiency and user control of what's being transferred, as this feature is meant for very frequent use. Only the contents of the two folders that are shown in the active/inactive panes are considered each time. Still, subfolder synchronisation can be assisted by the **mirror browsing** feature mentioned earlier.

## Internal Editor/Viewer

With 2xExplorer, the contents of each file can be examined without resorting to the default associated application, by either viewing or editing it as "text" (naturally the double-click mechanism is still supported). This feature in a sense provides a second "association" for each file. Since the editor is embedded in 2xExplorer, it provides access to the file contents much faster than e.g. using Notepad. The Notepad-beating capabilities are not limited to speed though. It features drag/drop editing, word wrapping, printing, unlimited size editing, plus it sports a handy toolbar for frequently used commands.

Although it won't substitute your main editor (say for writing your source code), it is nevertheless a very compact quick'n'dirty utility to be appreciated by programmers and lay users alike. The key objective is efficient everyday use, facilitated by the following main features:

- Keyboard shortcuts for most commands
- Powerful search and replace, including command history and *cyclic* search
- Font selection, colour customisation and tab settings
- Hexadecimal viewer for binary files - for the hacker within oneself
- Rich text format (RTF) viewer in read only mode
- Autotext for easing the insertion of opening/closing HTML tags and other text.

Especially for programmers of all shapes and languages, the following added items would surely prove handy:

- Indentation control
- Brace matching, including HTML angle brackets <>
- Goto line # command
- Line/Column indicator
- Insertion caret bookmark

The internal editor is also available in standalone version, for people that want to permanently associate all text files with it. Just visit the website and download **2xEditor**.

## Integrated Find Files Command

The standard windows Find Files utility is just beginning to catch up with the functionality offered by the 2xExplorer's internal item - and not in all departments. Still, you don't have to splash out for the latest version of Windows 2000 to have a decent file search utility, when 2xExplorer provides all these features for all earlier win32 platforms (9x/NT4 etc), too:

- Multiple file filters (file types) to locate.
- Many search "root" folders can be specified to look in.
- Date-constrained searches can use all three pieces of file date stamp information (modified, created and accessed). The minimum timeslice supported is an *hour*, which may help pinpoint recently changed files.
- Text-constrained searches can be refined by options like case sensitivity, word search, and the *time-saving* option to exclude binary files (regardless of exact type) from the searched list.

The find results window is a modeless dialog, which means that it can be left open without interfering with the normal operations in 2xExplorer. The main purpose served is to load and highlight the items found in one of the regular 2xExplorer panes, where any further operations may take place if need be. The range of supported commands in the results window itself is limited, as accessible via a context menu. In this sense it behaves differently to the standard windows Find Files utility; if you prefer the latter you can invoke it from within 2xExplorer using the shell context menu (right-clicking) on any folder item in the two panes or tree view.

The search results can be printed using the context menu.

## Advanced Features

2xExplorer was developed for non-frustrating everyday use in mind. Following is a list of features that will be appreciated under this light, by all people poised with seeking maximum efficiency and optimization in their lives - some jealous and/or incompetent folk would maliciously use the term "control freaks" here. :¬)

### Efficient Folder Navigation

There are so many different ways to move around your filesystem hierarchy that you are truly spoiled for choice:

- Using the address bar to key in paths, or to select a folder browsed earlier from the drop-down history portion. Buttons/shortcuts for moving to parent/root/desktop folders will also be found here.
- Dragging folders or files from your desktop and dropping them on the address bar is equivalent to typing their full path, and takes you directly to their location.
- Using the tree view to select any folder in all the filesystem hierarchy.
- Using the DriveBar for quickly accessing all disc drives installed on your computer (a similar functionality is available via the Tools | Select Drive command).
- Using the navigation chain, which allows moving to folders browsed earlier/after in the familiar Previous/Next mechanism popular with internet browsers. 2xExplorer also remembers the last item that had the focus and *reinstates* it when a folder is revisited.
- Using the bookmarks, which are shared by both panes.
- Many modeless dialogs that show filesystem trees (e.g. Folder Data, Find Files results) can also be used to change folder by double-clicking.

Since there is just one set of controls (e.g. tree, driveBar, miscellaneous buttons, etc.) shared among the two panes, any navigation command will normally affect the *active* pane. However, by holding down the **`<Alt>`** key the command will be directed to the inactive pane. This is a very useful feature which importance for improving browsing efficiency cannot be overstated enough. The `<Alt>` modifier works universally for all browsing commands mentioned above, regardless if they are keyboard or mouse-oriented. For example, you can hold `<Alt>` while double-clicking on a folder displayed in the tree view, while pressing `<Return>` in the address bar, while using `<Backspace>` to move to the parent directory, and so on.

## Multi-purpose Address bar

The address bar is multi-functional, accepting pathnames, visual filters or both at the same time. This is a handy feature allowing a single focal point to be used for all text-input requirements, doing away with the need for separate time-consuming sessions in dedicated dialog boxes that are found in rival file management applications.

Concentrating on the addressing functionality, either absolute or relative paths are accepted, including any combinations of the parent (`".."`) and root (`"\"`) folder specifiers. In many occasions addressing a folder by its name is much faster than furiously clicking here and there, especially if the target folder is more than a single step away in the hierarchy tree. Finally, it should be noted that all the boxes that accept paths in 2xExplorer (throughout the application, as e.g. in the Copy dialog), have similar "smart" parsers supporting them, and are fully path-"conscious", both for relative and absolute paths. For instance the difference between `"C:"` and `"C:\"` is properly resolved. **UNC** paths are also supported, allowing direct browsing of resources in other computers in a network (e.g. `"\\share\~umeca74"`). Finally, it is possible to include a filename as a part of the path, which results in browsing the folder and selecting the filename specified at the same time.

Except for addressing, the bar can also be used for entering commands to **execute**, for both DOS and 32-bit windows programs. In order to differentiate command execution from normal addressing, 2xExplorer interprets as commands inputs that begin with '`$`' or '`>`' characters. 2xExplorer also simplifies the compilation of argument lists, since it enables copying filenames from the active pane directly into the address bar, using `<Ctrl+arrowKeys>` to select names and `<Ctrl+Return>` to transfer them. Path names are easily accessible with the $L and $R constructs, for the left and right pane respectively, as is the focused filename with $N (similar to the GenerateBatch command, see below).

The drop-down portion of the folder address selection combo box lists the names of the folders that have been recently visited. This allows rapid switching backwards and forwards between folders, if need be. This list is different from the navigation chains maintained by each pane, since the latter are specific to each pane whereas the former is a shared resource.

**Histories** of other items entered in the address bar, like visual filters and commands, are also maintained. Naturally, there's no room for them in the drop-down portion; access is offered via items in the Files and Tools menus, respectively.

### Smart Keyboard Shortcuts

2xExplorer is also known as the "keyboard-junkie's"-Explorer. A design specification of the program is to provide keyboard shortcuts for all important and commonly used commands. Most accelerators are shown next to the respective menu command and are also included in the tooltips, acting as a quick reminder. For many useful "undocumented" shortcuts please refer to the accompanying `Keyboard.txt` file.

2xExplorer caters for the needs of a sizeable portion of computer users who prefer the keyboard for applications they use daily. With long term use it is easy to memorise the keyboard equivalents of most frequently used commands, which enable one to do things with lightning speed compared to using the mouse - that's the typical "man or mouse" dilemma. Why bother with target practising or do battles with clogged up and foul mouse roller balls, when paradise is just an arm's length away? 2xExplorer won't test your patience when you use it daily, even after the initial impressions have been long since worn off. That's the recipe for healthy long-term "relationships" IMHO.

Another useful attribute here is that the `<Ctrl>` key can be used to modify the behaviour of an accelerated command. For instance, the `<F5>` key used for copying will normally summon a dialog box, requesting the target folder for the operation. Although the "opposite" pane folder name will be usually pre-selected, and the only action required from the user would be a single Return key press for acceptance, wouldn't it be better if one could do away with the dialog box altogether? That is exactly what the Control key does as a modifier to commands like Copy: pressing `<Ctrl+F5>` will result in an *immediate* copy, dispensing with the dialog. All commands that use an intermediate dialog box and could somehow accept a default answer without any user intervention can be similarly modified using the Control key. For full details refer to `Keyboard.txt`.

### Folder Subtree Data

Inquisitive minds will surely find the summary information offered by a standard property page of any folder to be rather on the brief and utterly unsatisfying side. 2xExplorer provides detailed information about a folder including detailed breakdown of the included files and sub-folders (and the respective storage space occupied) all the way down to the last sub-folder in the hierarchy tree.

The results are presented in an intuitive tree fashion, hosted by a modeless (and resizable) dialog window, which can remain open for as long as one likes. That will help you to figure out exactly how on earth all these megaBytes managed to pile up in the windows directory.

Right-clicking on this window opens a context menu which may be used to sort the items and even print the fully expanded tree of information.

### Extended File Date Information

All 3 pieces of file date/time information available (last Modified, first Created and last Accessed) are available in the details view - by selecting the relevant property from View | Options page - unlike the Explorer where only the last modified time can be seen. This feature, except for allowing easy access to file time stamp information, also enables the sorting of files using dates other than the modification one. This is useful in many cases where one would like to detect e.g. files that have been recently accessed (without being modified). Those nasty virus and/or install programs will no longer hide their activities from power users!

### Batch File Creation

Except for the regular operations on selected files/folders, there is a special treat for all the old hands out there, that most probably would have wished for a similar capability all those years back in the dark DOS ages. Enter the batch file generator. Using a command template with special characters for the

path/filename/extension, it is possible to apply the same series of commands on each selected item with minimum effort. The following special characters are supported in templates:
**$N** for full filename (e.g. `object.cpp`)
**$B** for base filename (e.g. `object`)
**$E** for filename extension (e.g. `cpp`)
**$L** for left pane full path name
**$R** for right pane full path name
**$xx** for an automatically incremented number (see explanation below)

Typical uses of this command include the **mass-renaming** of file extensions (using a template as "`ren $N $B.other`"), touching the date stamp information, and so forth. You'll surely regret deleting them ms/dos commands!

Mass-renaming is made easier by using counters, which are numbers automatically incremented by one for each line generated. Name sequences like `file01, file02,`… are easy to construct with the special template "`file$01`". You have the flexibility of specifying both the starting value of a counter and the number of leading zeros; for instance, the counter "`$0012`" will start counting from 12 and add enough leading zeros to ensure a 4-digit number.

## Command Line Arguments

2xExplorer can be started with optional arguments (e.g. using a desktop shortcut). The command line can be used to override various default program settings, and markedly so the initial directories displayed in its panes. The syntax is as follows:

```
> 2xExplorer [leftFolder [rightFolder]] [/S:<name>]
```

All arguments are optional. The initial folder arguments can be utilised to make 2xExplorer the default file manager of your system; preferably you should specify full pathnames here. The `/S` option controls the registry key name where all the persistent program properties are kept (window size and the rest). By default, 2xExplorer will store everything in "`HKEY_CURRENT_USER\Software\ZabaraKatranemia Plc\2xExplorer`"; using the (for example) `/S:OTHER` argument will force properties to be stored/read from the "`HKEY_CURRENT_USER\Software\ZabaraKatranemia Plc\2xExplorerOTHER`" key instead. This feature can be handy for keeping various shortcuts in your desktop, each with different settings.

## Layout Management

2xExplorer uses advanced layout management to arrange the numerous toolbars that provide useful services. All bars can be either docked or floating, or even hidden if not required. Moving toolbars around is simple as long as you find an "empty" spot to drag them from. Holding `<Ctrl>` pressed will force a toolbar to float, so that it can be placed outside the main 2xExplorer frame window and/or resized. Special bars like the folder tree can be resized even while docked. The management algorithm permits having more than one bar sharing the same frame window edge while docked.

Given the large number of possible toolbar arrangements, 2xExplorer offers a mechanism to "save" a number of favourite distinct layouts and re-apply them at will at a later stage. For instance you may have an image previewing "mode", where the thumb control has a prominent position and one regular pane is hidden, another for regular file management where the window is maximised, and so on. The layout manager allows rapid switching among such different functionality modes. The information retained includes: window size/position, toolbar visibility/dock state/size/position, left/right pane sizes, even the view style of the panes (details, large icons, etc.).

## General Usability

The author has pledged for the utmost usability of this program. All dialog boxes offer reasonable suggestions so that you won't have to do much typing yourself. Even in case you do need a bit of typing, the most appropriate field will be usually pre-selected so one can start altering the information immediately and mouse-free. Another anti-typing measure is the maintenance of **history** items, so there is no need to re-type what you just entered a short while ago. As a matter of course, most boxes meant for keyboard editing will have a drop-down part carrying the recent history information (combo boxes).

In order to make accessing previously typed information even easier, all combo boxes support **autocompletion**, which comes in handy when there are many strings in the history list. Just type in a few characters of the intended text, hit `<up-arrow>` and 2xExplorer will expand the string for you, searching the history list for the combo box in question. See Keyboard.txt for the full instructions.

All this history information is saved (in the registry) when you quit 2xExplorer, and is made available next time you restart the program. This holds true for all the program settings. Properties that affect the program operation (e.g. the overwrite confirmation in the Copy/Move dialog box), window sizes and positions, even the width of each column in the "Details" view, are **persistent** properties. So each time you restart 2xExplorer it's like you pick it up where you last left it - no messing about with resetting your favourite options.

Finally, 2xExplorer makes good use of the status bar to communicate various important messages to the user, avoiding intrusive message boxes that require user intervention to get them out of the way. These status messages will clear themselves after 5 seconds or so without giving you any hassle. That's exactly the fundamental concept behind 2xExplorer: a set of simple yet important touches that make one's long term relationship with computers less of an irritant and more enjoyable.

*So, go on, let 2xExplorer make you more productive than thy neighbour!*

## Credits

2xExplorer was developed using Visual C++ (utilising MFC classes), single-handedly by **Nikos Bozinis** in 1999-2001.

2xExplorer website: http://personal-pages.ps.ic.ac.uk/~umeca74/

*Thanks out to (alpha-numerically sorted):*

- Robert Helmer, for all his assistance spreading 2xExplorer on the net.
- Dennis Savidis, for UNICODE advice that enabled 2xExplorer to run on windows NT platforms.
- Oliver Smith, whose CCoolDialogBar class provided the foundation for the dockable/resizable toolbar that hosts the filesystem tree view.
- Christian Storgaard, for coming up with that pukka icon design for 2xExplorer.