FindExe V1.12 - Contents

Note:

This document contains several references to the Windows SDK (colored red). If you wish to use them, but the help file cannot be opened, you can do one of the following:

1 Open "WIN31WH.HLP" manually using the "File-Open"- Menu and return to this document with **"Back".**

2 Ensure that the file can be found by one of the ways described in search rules.

Controls

General:

- "Find"- Edit Box
- "As File"- / "In Memory"- Switches
- "Module List..."- Button (only "In Memory")
- "DLL" / "EXE" / "*"- Switches (only "As File")
- "<" / ">"- (Back/Forth-) Buttons
- "About"- Button
- "Search"- Button
- "Close"- Button

Directories:

- General Considerations on Directories
- Selecting a Directory
- "Current"- Combo Box
- "Windows"- Display Field
- "Windows System"- Display Field
- "Program"- Combo Box
- "PATH"- Edit Box

Results:

- "Results"- Area...
- "Load"- Button
- "Unload"- Button
- "Rename"- Button
- "Delete"- Button

Additionals (System Menu + Accelerator):

- Touch   (Ctrl+T)
- Remove from memory   (Ctrl+Delete)

The "Find" Edit Box

You enter the search term in the **"Find"-Edit Box**. You can specify a file name, a part of a file name or the part of a file name using wildcards - depending on what you want to do (you *cannot* specify a path - see **note** below):

Using the Search Method "In Memory"

you can just specify parts of a name. **FindExe** then searches for modules, whose names *begin* as specified. You need not provide an extension for this method. If you do, it will be ignored(!). Also wildcards are ignored - to be more precise: the search term is cut at the first occurance of a wildcard. Since always a "begins- with- search" is performed, this is almost no restriction in practise.

But you need not specify anything: in this case pressing the "Search"- Button will find the very first module and using the "<" / ">"- Buttons will take you back and forth through the list of loaded modules.

Using the Search Method "As File"

wildcards are evaluated as usal: the asterisk "*" is a placeholder for a random count of characters, the question mark represents exactly one character.

Using this method you can furthermore specify any file extension. For convenience you may use the "DLL" / "EXE" / "*"- Switches to select one of these most-needed types. These switches are disabled, if you specify an extension in the **"Find"-Edit Field**.

Note:

Please note, that you can enter random text with (almost) random length into this field, but only a file name in its maximum length is evaluated. So you cannot enter any path specification. If you do, this will probably generate confusing error messages.

The "As File" / "In Memory" Switches

These switches determine the **Search Method**, which in turn determines, if the search applies to physical files on physical media (hard drive / floppy disk) (Search Method "As File") or to a module in Windows' internal list of loaded modules (Search Method "In Memory").

The "Module List..."- Button

The **Module List**, which opens, when you press the **"Module List..."-Button**, grants an "inside-view" at all currently loaded modules. The file name is placed at the beginning of each list element to enable you to jump to files beginning with a character that you type on the keyboard - for most times you will know the filename rather than the module name or the number where to jump to... Therefore the number appears in the second column.

Tip: You can open this list though the **"Module List..."-Button** may not be visible: press **F6** or select **"To Module List"** from the system menu.

Restricted / non-restricted display:

The current search pattern, that you (possibly) have entered into the "Find"- Edit Field, serves as a restriction, that means as a "filter" for modules to be displayed in the **Module List**. Once the **Module List** dialog has opened, you can choose between "all" or "restricted" display using the **"*.*"-Switch**. If there is no restriction, this switch is disabled.

This restriction again is applied to both: the file name and the module name of the module. This means, that the module is displayed, if either its file name *or* its module name begins with the search word (see also Search Method "In Memory").

You can still change the restriction after opening the **Module List** - you do not need to close it to accomplish this:

> Change the focus to the main dialog and enter a new search pattern into the "Find"- Edit Field (e.g. "M" oder "M*" to see all modules beginning with 'M') and click the **"Module List..."-Button** (or press **F6**) again. If you only activate the Module List dialog instead (eg. by mouse click, Alt+TAB etc.), this will not have the desired effect.

Updating the List:

The **Module List** is updated each time the window becomes the active window to reflect the actual state of loaded modules (therefore the list may "flicker" something, because the whole list is destroyed and created again and the selection bar must be repositioned).

If you are in doubt of the list being up-to-date*****, simply press **F5**. This will enforce an update.

The main dialog of **FindExe** sends messages to the **Module List**, when a new module is loaded or a loaded one is unloaded. Because of this the **Module List** is also updated in this case. It *cannot* reflect the actual state, if modules are loaded or unloaded by other means than by **FindExe** itself. In this case you must make the **Module List** the active window manually (or press **F5**) to enforce an update of the list.

This also applies to possibly further instances of **FindExe** running in the system, since there is no communication between different instances of **FindExe.** (Starting with V1.11 this usually is not possible, since the main dialog is just brought to the foreground, if you try to start **FindExe** a second time. But you can "fool" the program by copying it, renaming it and starting it with the changed name.)

***** Sometimes you may experience, that the module list is not up to date, e.g. when the window is not activated explicitely, but automatically e.g. by termination of another application.

Deleting/Unloading Modules while in the Module List:

You can use the **Delete** key to unload modules without changing to the main dialog. The **Module List** then switches to the main dialog where you can have a look at all data belonging to the object before you actually decide to unload it. Afterwards the main dialog switches back to the **Module List**, where you have come from.

Exiting the Module List:

"OK" (or **Return / Enter**) selects the currently selected module for display in the main dialog and closes the **Module List**.

"Cancel" (or **ESC**) only close the list.

Further Hints on Control:

Please have a look into the system menu of the window. There you will see all available options and the keys associated with them. Following a short description:

Navigation / Operation Keys:

You can use the following keys for navigation in and operation of the list (keys with an additional description are also available via the system menu or as a button):

F4 - "Sorted"

Toggles sort mode. Usually the list is displayed in an unsorted state. Hereby you can switch to alphabetic sorting to find modules more easily by their file names. Pressing **F4** another time switches back to the usual unsorted mode.

F5 - "Update List"

The list is updated / refreshed in any case, if you press this key. If you are in doubt, if the list is up-to-date, simply press this key.

F6 - "To Main Dialog"

Brings the **FindExe**- main dialog to the top.

F7 - "Continuous Update"

Switches to "Continuous Update"- mode. This means, that for each newly selected element of the list, the main dialog displays its data in the "Results"- Area. To actually see the results, it is best to put the module list aside by **F8** (see below).

Tip: If you wish to see the data of a module without closing the module list (**"OK"**), then use **F7**. If you don't like the "Continuous Update"- mode, because it takes more time, then simply press this key once again to switch back to normal mode.

F8 - "Change Position"

Switch between 2 display modes. "Mode" only means two different positions / measures, which are stored for the dialog during runtime of **FindExe** and which you can freely resize and reposition:
Display mode 1 is initially centered to the screen dimensions and is usually wide enough to show all available properties.
Display mode 2 is initially positioned to the right and is less wide, but higher and thus can show more entries at a time.

'*' (asterisk key) - "All / Restricted"

This key works the same as the **"*.*"- Switch** and thus is only active, when this switch is enabled. It makes no difference if you use the key of the numeric pad or the "normal" key - with or without **Shift** (the latter only applies to the German keyboard layout. In US layout you can use the apostrophe- key - I did not test further layouts).

Home

Jump to the beginning horizontally. If you already were there or the list is wide enough to show all information, (that is: no horizontal scrollbar exists), then jump to the first element.

End

Jump to the end horizontally. If you already were there or the list is wide enough to show all information, (that is: no horizontal scrollbar exists), then jump to the last element.

Ctrl+Home / Ctrl+End

Jump vertically to the beginning / end.

Page-Up / Page-Down

Scroll one page up / down vertically.

Ctrl+Page-Up / Ctrl+Page-Down

Scroll one "page" left / right horizontally.

Ctrl+Up / Ctrl+Down

Scroll up and down by one without affecting the selection bar.

Insert - "Load..."

Load the selected module once again (and increase "Usage Count"). See: "Load"-Button.

Delete - "Unload..."

Unload selected module (and decrease "Usage Count"). See: "Unload"- Button.

Ctrl+Delete - "Remove from Memory..."

Unload selected module until its "Usage Count" is zero. See: "Remove from Memory...".

Return / Enter - "OK"

Accept selected module for display in the main dialog and close the list.

Esc - "Cancel"

Cancel and close the list.

The "DLL" / "EXE" / "*" Switches

These switches enable you to easily choose the type of the file that you wish to search for. "DLL" is the default setting.

Alternatively you can enter any file extension manually in the "Find"- Edit Box. Then the **"DLL" / "EXE" / "*"-Switches** are disabled and ignored.

These switches only affect file search and therefore are only available, when the Search Method "As File" is selected.

"<" / ">" (Back/Forth) Buttons

These buttons enable you to "scroll" back and forth using the Search Method "As File", as well as searching "In Memory" - presuming that multiple files or modules match the search pattern, that you specified in the "Find"- Edit Box.

Please also pay attention to the corresponding **notes** in the mentioned search- method- topics.

When using these buttons, keep in mind that the order of modules in memory is chronological rather than alphabetical or matching any other sort criteria. The modules loaded first are also found and therefore displayed first (the "No."- Display Field indicates the order by which the modules were loaded).

This applies to files in a similar way, where the order is even more arbitrary caused by deletions and insertions of files.

"About" Button

This button invokes a dialog with a short description of **FindExe**, which in turn leads to this help document by clicking the **"More"**- button. There are 2 more dialogs providing useful "quick" information and tips for using **FindExe**. You can find the contents of these dialogs in a slightly different form in this document, too.

As an alternative to using this button you can use the **"Quick Help Key" F2**.

See also: Help...

"Search" Button

This button starts the search.

If you already had started a search and "scrolled" back and forth using the "Back / Forth"- Buttons before, then you can initiate a new "first- search" by pressing this button.

"Close" Button

This button simply terminates the program.

The Directory Fields - general Considerations on Directories

The **Directory Fields** are positioned according to the order, by which the directories are searched. The **Search Directories** listed in the PATH Environment Variable are displayed and can be edited in the "PATH"- Edit Box.

Selecting another "Current" Directory or "Program" Directory by one of the corresponding combo boxes changes the Current Directory (sometimes also called "Default Directory/Path" or "Working Directory") - possibly for other programs, too, when they are started or when they select new drives.

The meaning for **FindExe** is, that the current path of a drive must be retrieved when changing the drive to restore it when the drive is selected later again. Otherwise you would have always start selecting paths beginning at the root...

Therefore the current path is set at selection time respectively (in both combo boxes). Nevertheless **FindExe** always regards the path displayed in the "Current"- combo box as the current directory and thus the point where to begin to search.

The directories have no meaning for modules in memory, for their complete names are already known.

More information on the directory categories:

Current Directory
Windows Directory
Windows System Directory
Program Directory
PATH Variable

Selecting a Directory using the Path Combo Boxes

You can choose the **"Current" Directory** and the **"Program" Directory** with the "Current"- and the "Program"- Combo Boxes.

While at first glance the behaviour of those combo boxes may appear somewhat unaccustomed, it is more convenient than the usual selection with double clicks:

A simple mouse click selects and opens a directory or a drive. You can open the list by **Mouse Click**, **F4**, **Alt+Up/Down** or simply the **Spacebar** or the **Down** or **Right** keys.

Note:

The "**..**"- entry is displayed in a network environment too, independently of the **"SHOW DOTS = ON"**- Entry in **NET.CFG**. Earlier Versions of **FindExe** required you to specify it in your **NET.CFG** file. This is no longer required.

Keys for operating the combination fields

Scroll Lock

This key switches between two display modes, which differ in how they position the **"[..]"**- entry. This is meant to simplify control via the arrow keys.

If the key status is OFF, the old mode applies: the "[..]"- entry - if existing - is positioned below the current selection, this means: always as the second element in the list. This mode can be advantageous, if you often need to branch back.

If the key status is ON, the new mode applies: the "[..]"- entry - if existing - is always positioned to the top. This mode is advantageous, if you need to branch into subdirectories more often than branching back. Then you need not skip the "[..]"- entry first. However: branching back via arrow keys is a bit more difficult in this mode.

Since the key state is inquired at startup time, the setting is still in effect, if you restart **FindExe**.

Left

Close the list (= "Cancel", this is identical to **ESC**, see below).

Right

Open the list or - if already open - select the current element. This is identical to **Enter** with the list open (see below).

Up

Move up through the list.

Down

Move down through the list. If the list was not open it is opened now.

Backspace

Move one level back/up - this has the same effect as selecting **"[..]"**. If the list was not opened, this key first opens the list and then branches one level back.

Control+Backspace

Branch back up to the root directory of the drive. Instead you may hold down the **Control** key while selecting the **"[..]"-** entry (via **Enter** or **Mouse**). You can use this key combination to simultaneously open the list, too (see **"Backspace"**).

F4

In case that the list is open, close it and keep the current selection. Otherwise open it.

Return / Enter / Alt+Down / Alt+Up / Spacebar

Open the currently highlighted directory or drive. If this element was the first one, the list is closed keeping the element as the current selection. Otherwise the list stays open for further operation. All these keys except **Enter** also serve the purpose of opening the list.

ESC

Restore the path that was selected before opening the list and close the list.

Mouse Control

Click the **Left Mouse Button** anywhere *outside* the list to close the list confirming the current selection. This has the same effect as **F4** (see above).

Click the **Right Mouse Button** anywhere *outside* the list to close the list cancelling the current selection. This has the same effect as **ESC** (see above).

"Current" Combo Box - the Current Directory

The **current directory** serves as the starting point for the search. At times this directory is also called **"default path / default directory"**, **"working directory"** etc. It is the path out of which the current operation is started. Furthermore it can be changed by several influences during program execution.

Out of these reasons I prefer to call it the **"current"** directory.

Do not confuse this with the "program directory":

If a program is started out of the **current directory,** both are identical at start time, but this may change during runtime (see above). At any rate they differ, when the application was startet using a path specification and this path is not the current path.

Examples:

- in the "file execute" box of File Manager you enter "C:\BIN\FINDEXE.EXE" instead of the pure file name "FINDEXE.EXE", while the current path is C:\DOS or
- you intentionally redirected the "working directory" in Program Manager.

See also: Selecting a Directory.

"Windows" Display Field - the Windows Directory

The **Windows Directory** is the directory, where WIN.COM and the GRP-, INI- and further user specific files reside. Hence this is the **"personal" Windows Directory** (in contrast to the Windows System Directory).

Internally it is retrieved by the SDK function

GetWindowsDirectory()

Subsequent to the Current Directory this directory holds the highest priority level.

In the dialog this is a pure display field.

"Windows System" Display Field - the Windows System Directory

The **Windows System Directory** is the directory, where the Windows- System- files (e.g. KRNLx86.EXE, USER.EXE, GDI.EXE etc.) reside. This is the shared directory, if Windows is used as a networked installation, otherwise it is the SYSTEM- sub- directory in the "personal" Windows directory.

This directory is searched after a file could neither be found in the current directory nor in the Windows directory.

Internally it is determined by the SDK function

GetSystemDirectory()

In the dialog this is a mere display field.

"Program" Combo Box - the Program Directory

The **Program Directory** is the directory that contains the executable file for the current application. This is - in contrast to the current directory - *always* the path from which the program was startet - more strictly speaking: *where* the program was found - no matter how. If this directory was just the current one or the program was found by a path in the PATH variable - all this does not matter.

The **Program Directory** is searched, after searching the current directory, the Windows Directory and the Windows- System- Directory did not succeed, hence holds the lowest priority except from the search directories.

Internally to a program it can be determined by retrieving the module name of the application itself via the Windows SDK function

GetModuleFileName()

and extracting the pathname.

The **"Program"-Combo Box** of **FindExe** provides the selection of this directory to enable you to simulate a **Program Directory** of your choice. For there is little use to have **FindExe** always search its own **Program Directory**.

If you do not specify a parameter for this directory, the default is **FindExe**'s own **Program Directory**.

See also:

Passing as a parameter:
Passing Parameters

Selection:
Selecting a Directory

"PATH" Edit Box - the PATH Variable / Variable List

The **PATH variable** contains the search directories, that are searched, if a program or a DLL could not be found by some other means.

The **Search Directories** hold the lowest priority level and are searched only, if the search in all other directory categories ("Current", "Windows", "Windows System" and "Program") has failed.

When **FindExe** starts, the **PATH-Edit-Box** displays the current contents of this variable. You can also edit it - the changed path will then be used instead.

Note that changing the contents of the edit box does *not* change the original variable. The scope is limited to **FindExe**.

Variable List

The **"?"-** button (or **F5**) opens a list of all environment variables, from which you can select any other variable and transfer its contents into the edit box. E.g. you might use this feature to find out where a particular C header file is found via the INCLUDE variable.

For more information on the operation of this list see the chapter Module List.

You can see additional functions and the assigned keys, if you open the system menu. Please see Module List for a more in- depth- description.

The "Results" Area

The **"Results"-Area** displays the results of the search. Most of the fields here are mere display fields. Only these fields are described in this chapter. You can see the description of the controls by selecting the links in the following text or by selecting the topics via Contents --> Controls.

The Display Fields (from top left to bottom right):

The State "LOADED" or "NOT loaded"

is to be seen in the upper left corner. **"No."** and **"Usage Count"** (see below) of a file that is not loaded always equal zero, for a *loaded* one they are always greater than zero.

"Usage Count"

to the right of the state field indicates, how often a file has been "loaded". This count does not mean that the file is actually in memory so many times, but that it is referenced and used by this count of applications. Therefore this is called "Usage Count" - you might also call it "Reference Count".

You can influence this count by the "Load"- and "Unload"- Buttons (see below).

Buttons for special actions on the file:

Beside the fields mentioned above you may or may not see one or more of these buttons - depending on the search results. Please select the reference(s) of interest for details:
"Load", "Unload", "Rename" and "Delete".

The File Name Area

begins in the second line and consists of two lines. In case that the name is too long to fit onto one line, it is wrapped. The path is displayed in lower case to save display-space, the file name is displayed in upper case.

The File Properties

appear below the file name, in case a file was found: size, date and time. The date is displayed according to the settings of "short date" in the control panel. Time is always displayed in the format HH:MM:SS (hours:minutes:seconds).

"No."

is displayed in the last line. This number represents the "time" when a module was loaded - of course it does not display the time, but the ordinal number.

In the "Category"- Field

you can see at one glance, by which means the module or file was found:

CURRENT	the file was found in the current directory
WINDOWS	the file was found in the Windows directory
WINSYS	the file was found in the Windows system directory
PROGRAM	the file was found in the "Program"- directory
MEMORY	the module was found in memory. This is always the case, when you use the Search Method "In Memory"

The "Module Name"- Field

last not least, displays the *module name* of the module, in case that it could be found in memory, otherwise "- -" for "no name available".

A module name consists of a maximum of 8 characters and does not contain a file extension. This is the name that was specified by the developer of the module in the module definition file by the "NAME"- statement (applications) or the "LIBRARY"- statement (dynamically loadable libraries, which is to say: DLL's), and is in most cases identical to the file name of the module apart from the file name extension.

Note that this name is always displayed when using the search method "In Memory". When using the search method "As File" it is only known and displayed, if the **"Usage Count"** field displays a value greater than zero.

"Load"- Button

This button enables you to load a file, if it is a DLL or execute it, if it is an application.

If both cases do not apply, this function has no effect, that is: Windows issues a message telling you that this file is not accessible.

This button is only visible, if a file was found - be it by file search, be it by accessing the file by the full name of a loaded module. If the file could be successfully loaded, the "Usage Count" increases by 1 each time you click this button. If this does not happen, an error occured when trying to load the file.

First **FindExe** tries to load the file as a DLL by

LoadLibrary()

If this does not succeed **FindExe** tries to execute the file via

WinExec().

If this does not succeed either, Windows issues a message (see above).

Before loading the module, **FindExe** sets the path displayed by the "Current"- combo box as the current path to set this as the working directory for the loaded module.

Please note, that the *"Program"- combo box* has no effect at all on a module being loaded by this method. Its **program directory** is its *own* path.

If **FindExe** realises, that the module has already been loaded via another path specification, it issues a message and you can decide, if you wish to switch to the loaded module to see its data. **FindExe** must switch to the "In Memory" mode for this and does *not* load the module to present its *current* state to you.

Therefore - if desired - you must initiate **"Load"** a second time.

This mechanism ensures the correct update and display of the "Usage Count" of the file. Otherwise you would indeed load the module, but not see the correct usage count, since the file (strictly speaking - that means with the same path spec.) *is* not loaded.

But Windows always looks for the module in memory first - not considering different path specifications. If it finds the module, no file with the same name is loaded anymore - only the usage count of the file already loaded is increased.

Notes:

1 As a result of FindExe's comparing the *file* names (the *module* name of a file is not known until it is loaded), it is not able to recognize that a module is loaded, if you try to load a renamed version of this file.

Example: MYDLL.DLL is loaded. The same file exists in the same or a different path with the name MY_DLL.DL_ and you choose "Load" for this file - FindExe does not recognize, that this module is already loaded - but the usage count of **MYDLL.DLL(!)** is increased.

2 The action is performed *immediately* - therefore the button name does not contain "...". In case that you accidentally loaded a file, unload it (repeatedly) until the "Usage Count" is restored to its original value.

If you are not sure - leave it alone! It is not critical, if a module is loaded too often (at least in most cases) - it usually only stays unnessecarily in memory even if it is no more needed (after restarting Windows this problem is solved in any case). But it is *utmost* critical to unload a module that is still needed by other modules...

See also:

"Unload"- Button and **NOTES** there.

"Unload" Button

This button enables you to remove a module from memory. This is useful, if an application crashed, leaving one or more DLL's in memory, which, of course, should no more reside there.

During software development sometimes you may wonder why an error - already corrected thrice - did not vanish. The solution might be: the DLL is still in memory from an earlier crash and therefore not loaded anew.

I know - it sounds illogical, for something like this should not happen - the DLL should be locked and thus prevented from being overwritten. Actually it *does* happen occasionally, especially when the DLL had been loaded from a different path and maybe out of further reasons... - also you may not be able - out of "mysterious reasons" - to set breakpoints in the debugger and more of that kind. So have a look using **FindExe** and sometimes you will be suprised...

More likely in most cases you will experience that you cannot relink the file and/or copy it to its destination, because it is obstinately locked by the system - which considers the file to be still "used".

Another "popular problem" is, that your program is immediately stopped with a "general protection fault" at startup time, because DLL's used by the program did not unload and reload correctly. Then e.g. the DLL tries to work with file handles that belonged to the task already stopped and thus tries to access invalid memory areas.

Before unloading the module **FindExe** asks you, if you are sure to do so. If you answer "yes" and the module could be successfully unloaded, then you can see the "Usage Count" decrease by 1.

If the module only was loaded once, there are two different cases:

1. "In Memory" is active:

The module is no longer loaded and can no more be the current module. **FindExe** automatically starts a new search for the *next* module that matches the search criteria. If there was only this match, the file name area ("Results"- Area) displays a message, indicating that a module could not be found.

2. "As File" is active:

"Usage Count" now is 0, nothing further happens.

In both cases the file should now be unlocked and you should be able to overwrite, rename or delete it (in a network it may still be locked by other users).

This function terminates applications by the TOOLHELP- function

TerminateApp(),

DLL's are unloaded by the SDK function

FreeLibrary().

The type of the module is determined by the following procedure:

GetModuleHandle()

returns the module handle for the file name. Then the task list is searched for a module using this handle using

TaskFirst()

and

TaskNext().

If there is no match, the module is a DLL, otherwise it is an application.

IMPORTANT NOTES:

BE VERY CAUTIOUS WHEN UNLOADING DLL'S!

Only unload modules, which you are absolutely sure are remainders of a previous crash of an application or are "hung" in memory by some other means!

ALWAYS UNLOAD MODULES "TOP/DOWN",

that is: if several modules are involved: do *not* unload the modules loaded by another DLL, but unload that DLL (which caused the load of the other module(s)). This DLL depends on the later-loaded modules. If you unload one of them, you risc the DLL to run into void... which is usually a "general protection fault" - or even *worse*....

Besides: when unloading a module Windows automatically removes any lower- level-modules that this DLL or EXE relies on (if they are no more used by other modules) - so *you* do not need to do so anyway - apart from exceptions - e.g. dynamically loaded DLL's.

For the purpose of recognizing the dependencies it may be helpful to have a look at the order of loaded modules using the Module List - *but do not rely on the order!* Sometimes low- level- DLL's are loaded earlier than their "high- level- partners"!

If you do not care about these warnings, you probably will experience bad crashes!

But even *if* you do: be aware that loading and unloading modules is an intervention in the Windows system handling which can lead to unpredictable results - even if carefully applied - not *necessarily*, but *possibly*.

Here "practice makes proficient" applies. In this case it means: usually you can tell more and more what goes and what may be riscful. At any rate it is good to keep these notes in mind.

Further Notes:

1 This button is only visible, if a module was found - be it by "In Memory"- Search, be it by finding a module in memory that belongs to a file found by file search.

2 The system- menu- entry belonging to this function is only enabled, if you can use the **Delete** key for this function, that is: only if none of the edit fields is active, which need this key for the purpose of deleting characters. In this case please use the button.

"Rename" Button

This button enables you to rename a file found by file- search. This is useful, if you have found a file in a place, where it is no more appropiate, but you do not want to delete it.

A file selection dialog box opens, where you can specify another path and a new name or just another path or just a new name.

Note, that you cannot select a path on a different physical drive, for **FindExe** does not contain a copy function - it simply uses the rename / move function of the system.

Note:

This button is only visible, when a file was found and is not in use, that means: the **"Usage Count"** (see "Results"- Area) is zero. If this is the case, this only means, that *you* do not use the file. In a network it may still be locked by another user.

"Delete" Button

This button enables you to eliminate a file found by file- search. This is useful, if you found a file in a place, where it should not be (any more).

Note:

This button is only visible, if a file was found and is not in use, that means: the **"Usage Count"** (see "Results"- Area) is zero. If this is the case, this only means, that *you* do not use the file. In a network it may still be locked by another user.

Touch (Ctrl+T)

This "touch"- function enables you to change the file date and / or the file time.

As an alternative to calling this function by menu item or the accelerator keys **Ctrl-T** you can invoke it by simply clicking onto the file properties display field.

Once the "Touch"- dialog for entering the data has opened, you can edit each field or leave them alone. You can set one or both values to the current system value by deleting the respective field(s) and leaving it/them blank. Click "OK" to confirm your changes and the "Results"- Area will be updated immediately with the data read from the file, so that you see the actual, current data.

If you don't make changes and click "OK", this has the same effect as if you clicked "Cancel".

Notes:

- When you edit these fields you must usually enter all data. You can omit single items (e.g. date: day, month, year) only in certain cases, since the value(s) for them are set to zero or - if you confuse their order - may even be set to values that you would not have expected.

  However, if you do want to set, let's say, the time to zero, you just need to enter a single "0" or a random non- numeric character (e.g. a blank space) (recall: leaving the field blank inserts the *current* time).

  Likwise you can set minutes and seconds to zero: just enter the hour only.

  You can enter the year without thousands and hundreds, e.g. "84" expands to "1984" and "1" to "2001".

- You can only "touch" a file, whose data are displayed in the "Results"- Area. If no file is currently available and you try to invoke this function, **FindExe** issues a beep.
- The date and time appear in the edit fields in exactly the same format as they do in the "Results"- Area. You must enter the data in the same format.
- It does not matter, which *kind* of separators you use to separate items from one another (day from month etc.). It only matters *that* you separate them by at least one non- numeric character.

Remove from memory   (Ctrl+Delete)

This option works the same way as the "Unload"- Button - except that it unloads the module regardless of its usage count - and thus is even more "dangerous" than the unload option.

This function is only available via the system menus of the main dialog and the Module List or the key combination **Ctrl+Delete**.

Please see **"Unload"- Button** for more details and **important notes!**

DLL's needed / Compatibility

This chapter is meant to inform you about the DLL's that are needed by **FindExe** itself. This information is essential, because you *must not* unload these DLL's by **FindExe** (at least not until their usage count is zero), otherwise **FindExe** would crash - which may affect the whole system...

But first a few words about

Compatibility:

FindExe runs under Windows 3.1 and later. You also need the DLLs mentioned below, that are not marked as "optional".

Now it is certain: **Findexe** does *not* run under Windows 3.0 or at least it will behave very unusual and besides the DLLs needed do not belong directly to the system under this version.

Findexe *does run* under Windows 95. See also: Notes on Windows 95.

1. TOOLHELP.DLL:

FindExe needs this DLL to inquire the data from the Windows Module List. It belongs to the system and therefore usually resides in the Windows- System- Directory.

2. COMMDLG.DLL:

This DLL is needed for the file selection dialog ("Rename"). Starting with   Windows 3.1 it belongs to the system and therefore usually resides in the Windows- System- Directory.

2. CTL3D.DLL (optionally):

This file is responsible for the 3D- look of **FindExe**. If this file is not found, **FindExe** runs without 3D- look.

In case that this DLL is in your search PATH (usually in the Windows- System- Directory), but you want **FindExe** prevent from using it, press and hold down the **ESC** key while starting **FindExe.** It will then not use CTL3D.DLL.

This file usually belongs to the system too, but is an *extension*, that means: it is not part of Windows automatically. I do not provide it as a part of **FindExe**, since most likely you will already have it in your Windows- System- Directory. If not, there are many ways to get this file, for example via the WinShare- Forum on CompuServe.

But FindExe also runs *without* CTL3D.DLL (it only does not look so nice).

Notes:

- The 3D- window- frame- effects integrated in **FindExe** can - in rare occasions - conflict with CTL3D.DLL (older versions). Because of this you can disable this feature too: hold down the **F2**-key while starting **FindExe.**
  I also recommend, that you upgrade to the latest version of CTL3D.DLL.
- In case that you wish to disable both (CTL3D.DLL and 3D- frames), simply press and hold down **ESC** *and* **F2** while starting **FindExe.**
- If you experience the problem "Could not find function ... in CTL3D.DLL" you should upgrade to the latest version of CTL3D.DLL. Earlier versions do not support the "auto subclassing" feature used by **FindExe**.

  Tip:
  Because it may be hard to find out, which the "latest version" of this file is (file dates differ for identical files and are newer for older files and vice versa!), look at the file size: it must be about 20 KB (versus ca. 13 or 14 KB).

Passing Parameters, Pre-defining the Directory Combo Boxes

You can pass the Program Directory and/or the Current Directory as parameters when calling **FindExe** - the Program Directory as the first parameter.

In case that you only want to pass the Current Directory, but not the Program Directory, type a comma just before the path name (do not use blanks between them).

If you wish to pass the Program Directory only, just pass it as the only parameter. A second one is not required.

Examples:

C:\BIN\FINDEXE ,C:\DOS

> determines that C:\DOS be the Current Directory, the Program Directory is initially set to C:\BIN (the program directory of **FindExe**).

C:\BIN\FINDEXE C:\DOS

> determines that C:\DOS be the Program Directory, **FindExe** inquires the Current Directory at program start time from the system.

FINDEXE C:\BIN C:\DOS

> determines C:\BIN to be the Program Directory and C:\DOS the Current Directory.

This order is somewhat in contrast to the search order and the order of the fields in the dialog. I chose it yet, because there are many other ways to set the Current Directory (e.g. via the "working directory" of Program Manager or "File Execute" in file manager out of a specific path). Therefore passing this is often not needed.

The Program Directory, on the other hand, cannot be set in any other way than passing it to **FindExe** or selecting it manually in the "Program"- Combo Box.

Help / "Quick Help" / Context Sensitive Help

Help / Context Sensitive Help:

You can request help at any time by pressing **F1**. Then this help file opens - presuming that it resides in the *program directory (!)* of **FindExe**.

In the main dialog the help topic for the currently active control opens (context sensitive help).

In the **Module List** always the help topic for the Module List and in the **Variable List** always the help topic for the PATH- Edit Box / Variable List is displayed, if you request help.

Using **Ctrl+F1** and/or **Shift+F1** you can force the **"Contents"** page to be invoked.

"Click Help":

Starting with **V1.12** you can use the right mouse button to request context sensitive help. This enables you to get help quickly for both: areas that *can* and areas that *cannot* have the focus.

E.g. (right-) click anywhere into the "Results"- Area to open the help topic on "Results". Or try clicking the group boxes' frames / titles to get help on directories (**"Directories" group box**) or controls (**"Find" group box** - untitled) in general.

A right click on the icon or into any area that is not linked to a help topic will take you directly to the **"Contents"** topic.

"Quick Help":

For "quick help" 3 help dialogs are available explaining basics about the main dialog and search rules and giving tips on search. You get there by pressing **F2** or the "About"- Button.

Search Method "As File"

Using this search method you can search for a file specified in the "Find"- Edit Box according to the rules Windows searches for executable files (see also: Search Rules).

You can freely use the wildcards **"*"** and **"?"**. If you do so, you can search for the respective previous and next file using the Back / Forth- Buttons ("<"/">").

Backward- search searches along the PATH variable from right to left, then searches the Program Directory, then the Windows System Directory, then the Windows Directory and - last not least - the Current Directory. Furthermore in each directory the file list is searched from the last to the first entry - hence exactly the other way round.

If a file is found, **FindExe** also examines the internal module list, if there is a module matching the file. The module is only regarded as to match, if the *file name* (not the *module name*) of the module, including the file extension, matches the name of the file (the module name *cannot* be compared, since module names of files are not known to **FindExe**). See also: Search Method "In Memory".

All data gathered this way are displayed in the "Results"- Area.

NOTE on Backward-Search:

Please consider that for this kind of search the directories must be searched from the first entry up to the current entry to find the previous one. There is no system defined way of searching directories in reverse order.

This can (under certain circumstances) be a rather time consuming task and you may have to wait a while, before **FindExe** is back from its "previous- search".

The "circumstances" are: many entries in the PATH variable, many many file entries in some or all of the paths involved, several paths pointing to a network where there is heavy traffic...

But used with some care backward- search can also be useful in such situations.

Search Method "In Memory"

This search method searches for a module within the list of loaded modules, managed internally by Windows. You can view this list by clicking the "Module List..."- Button, which is visible, when **"In Memory"** is selected or by pressing **F6**.

A module is found, if either its module name or its file name match the search term. File name and module name need not be identical as you can see with the very first module loaded: its module name is "KERNEL", but its file name is "KRNL386.EXE" or "KRNL286.EXE".

The module name is the name specified by the developer of the program or library in the module definition file as "NAME" (EXE-files) or "LIBRARY" (DLL-files).

Since file name and module name can differ, you should have a look at both in case that you are confused about the search result.

Because a module name does not contain a file name extension, an extension is ignored when using this search method - in case you have entered one in the "Find"- Edit Box. Out of this reason the "DLL / EXE / *"- switches are not available in this mode.

For this search method a simplified wildcard handling applies:

> At the first occurrance of a **"*"** or a **"?"** the search word is cut and a "begins- with-search" is performed. Because thus *always* a "closest match" is searched, wildcards need not be specified. **FindExe's** tolerating wildcards, which are not needed, is helpful, if you must often switch to and fro between both search methods.
>
> **Tip:** Since you need no wildcards for "In Memory"- search you can begin to "scroll" through the module list back and/or forth using the Back / Forth- Buttons immediately after invoking FindExe or after deleting the contents of the "Find"- Edit Box.

If a module is found, **FindExe** tries to access the corresponding file to retrieve and display its data (size, date and time). The file name is retrieved by the TOOLHELP.DLL-functions of the Windows SDK

ModuleFirst()

and

ModuleNext().

All data gathered this way are then displayed in the "Results"- Area.

Related Topic:

Search Method "As File"

General - ***IMPORTANT for First-Time-Users***

This Software is FreeWare. This means, that you are allowed to use it without fee.

BUT READ THE FOLLOWING NOTES AND RECOMMENDATIONS!:

- YOU ARE ALLOWED TO PASS THIS SOFTWARE ON TO OTHER USERS, BUT ONLY ON A NON- COMMERCIAL BASIS AND ONLY UNDER THE CONDITION, THAT YOU INCLUDE ALL FILES BELONGING TO THE PACKAGE - ESPECIALLY THE ONE CONTAINING THIS INFORMATION.
- YOU ARE **NOT** ALLOWED TO APPLY ANY CHANGES TO THE SOFTWARE AND/OR ANY FILES ACCOMPANYING IT.

DISCLAIMER:

THE AUTHOR OF THE SOFTWARE (Dieter Hollmann) STATES HEREBY, THAT HE CANNOT AND WILL NOT ACCEPT ANY LIABILITY ARISING OUT OF DAMAGES THAT MIGHT RESULT FROM THE USE OF THIS SOFTWARE.

The progam is tested and operational. But it cannot be garantueed that it operates properly under any configuration and situation imaginable or non- imaginable. Also it may not be free of errors.

This applies to the documentation analogously: there is no guarantee in the accuracy of the information provided - despite all efforts were taken, errors may have crept in and / or parts may not be up-to-date due to development.

Due to the special character of this program it is highly user- dependent, how the software and the whole system behave when using this software. So read (at least) the recommended sections carefully!

IF YOU DO NOT AGREE TO ALL TERMS MENTIONED ABOVE, REMOVE THE SOFTWARE FROM ALL OF YOUR MEDIA, DO NOT USE IT AND DO NOT SPREAD IT!

As a first-time-user you should read at least the following sections:

IMPORTANT:

"Unload"- Button, above all: **NOTES** there
DLL's needed
General Notes on Usage and Operation

Recommended:

The "Find"- Edit Box
Search Method "As File"
Search Method "In Memory"
The Module List

General Notes on Usage and Operation

Use(age) of the program:
(especially Non- Programmers: please read this!)

This program is meant to be a tool for software development and therefore will be used, and should be used usually by program developers. This does not necessarily mean, that it is not usable by non- programmers at all. But as such take particular care in using the **"Load"-** and **"Unload"-** features!

You may then also need further or more basic literature to understand the documentation. Unfortunately I can make no recommendations on this, except the documentation of the Windows SDK. This in turn requires some basic knowledge in software development - a dilemma.

However: if you are careful, you are likely to have few problems, especially since the program warns and asks for confirmations before performing critical actions.
Remember that you act on your own risk (see Disclaimer)!

At any rate save files you are working on, which could be lost, if an action causes a system crash! It is not program errors of FindExe (at least no such errors are known), but problematical usage and/or the system already being unstable, that can lead to such problems.

Operation:

- Also have a look at the system menus of the windows **(ALT+Spacebar)**. There - below the standard functions - you will see all additional options and the accelerator keys assigned to them. Note also the additional keys for standard functions such as **Ctrl+F10** for **"Maximize" / "Restore"** and **F9** for **"Minimize"**.

- Now all windows (main dialog, module list and variable list) are resizable and appear in 3D- look. Do not bother: you *can* resize them, though, judging from their appearance, you may not expect it - this is the new "Windows95 style". Simply grasp the windows at their edges as usual.

  For the main dialog this is probably of little use. The main advantage is, that it can have a minimize button, indicating it is minimizable.

History - New Features and Corrections

V1.12	-	08-Aug-1995
V1.11	-	02-July-1995
V1.10	-	08-Jan-1995
V1.00	-	**03-Nov-1994**

History - V1.12

Problem Corrections:

- When started without CTL3D.DLL and the main dialog was iconized, all other actions / applications were blocked except from double clicks and Alt+TAB (sorry for that!).
- **ESC+F2** at startup (see "Notes" in DLL's needed) did not work exactly as intended (3D-frames were still drawn).
- When there is an error initializing CTL3D.DLL, 3D-Controls are now completely deactivated - there are no more error messages during run time after the initial message.
- Combo boxes: restoration of the previous path did not work correctly in some situations.
- Help file: corrected some language/keyword errors, added more key words.

New Functions / minor Corrections:

- Now a module can be removed from memory in one step, independently of its UsageCount.
- Context- sensitive help has been improved ("Click Help" via right mouse button).
- The main dialog now has a "Touch" function.
- When you activate **"Continuous Update"** (**F7**) in the Module List and the main dialog is minimized, it is now restored to its previous size.
- The main dialog is no longer brought to the top, when you close the Module List via "Cancel".
- Now the main dialog is "maximizable", too. This means, its original size and position can quickly be restored this way. This is comparable to (but not exactly the same as) the **"Change Position"** function of the Module List and the Variable List.
Furthermore this dialog can no more be given a size larger than its original size, thus making it easier to restore the original size by resizing the dialog.
- When loading or unloading modules in the main dialog, while the Module List is open, now the Module List sets the selection bar to the module affected by this action or - if this is no more loaded - to the module that follows or - if none follows - to the last module.
- Now you can use **F5** / **F6** to invoke / change dialogs even if the dialogs are minimized, but only if the popup- system- menus are not displaying - hence when you activate the icons via **Alt+ESC** or remove the system menu by clicking somewhere on the desktop.
- Combination boxes:
° Operation of the list via mouse and keyboard has been improved. There are additional keys and the meaning of some keys has slightly changed. See Selecting a Directory for details.
° The ".."- entry is now always displayed in network environments, independently of the "SHOW DOTS = ON" entry in NET.CFG. See Selecting a Directory.
° The ".."- entry is no more shown for the root of drives redirected by SUBST.
° Root directories are now displayed correctly with a terminating backslash.

---===*===---**

History - V1.11

- Now all windows (main dialog, Module List and Variable List) are resizable and appear in 3D-Look. See General Notes...
- The operation of the combo boxes and the Module List has been improved. See also: Selecting a Directory and **Navigation and Operation Keys** at the end of the chapter about the Module List.
- The date format for displaying file properties is now taken from the settings in WIN.INI which is equivalent to the settings of the control panel. Please note that the time is still *always* displayed in the format hh:mm:ss.
- Applications can now be started by pressing the "Load"- button. See also: "Load"- button.
- Additional keys now simplify control. All additional functions and keys are documented by the system menus and - of course - are selectable there.
- The PATH variable is now visible and can be edited. Furthermore you can select any environment variable into the PATH- edit box.
- Context- sensitive help is now available - see: Help...
- The Module List now is always available, not just during the search method "In Memory" More information: Module List.

---===*===---**

History - V1.10

- Instead of "As File" now "In Memory" is selected at startup, since this is needed most often.
- The "Module List..."- Button was implemented. This button replaces the "EXE / DLL / *"- switches during search method "In Memory", where this switches had no effect anyway.
- Corrected the handling after unloading a module. Now it works as documented in "Unload"- Button.
- At startup the "Find"- Edit Box now contains an asterisk "*". Thus you can immediately search for files ("As File") as well as for loaded modules ("In Memory") using the Back / Forth- Buttons without typing any character before.
- Before loading a new module **FindExe** now checks, if this module is already loaded via a different path specification and - if so - tells you. Formerly in such cases the usage count of the loaded module was increased continually, but you could not see it. More information: "Load"- Button.

---===*===---**

Notes on Windows95

Reading the following notes, consider that at the time of this release Windows95 was still a beta version. This information is mainly (but not only) given for beta-users.

FindExe is not a genuine Windows95- program, but runs without restrictions under this OS. However the 3D-controls integrated in Win95 do not take effect on **FindExe** (apart from common dialogs).

Therefore - if you prefer 3D-look - you must still use CTL3D.DLL. Unfortunately the common dialog boxes will have a "double 3D-effect" in this case. But this should not matter too much, since they are only needed in rare occasions.

Known Problems:

Because of various (harmless) problems with scrollbars (only under Win95) I use the LBS_DISABLENOSCROLL style for the list boxes. The effect visible to you is that the scrollbars are always displayed - wether needed or not.

Further problems are not known to me at this time, but may arise due to the beta state of Win95...

Suggestions, Problem Reports, Other

If you have suggestions, comments and/or problem reports, send an E-Mail via CompuServe to:

Dieter Hollmann
User-ID: 100410,3525

via Internet:

100410.3525@compuserve.com

If you have no opportunity to send E-Mail maybe a friend or acquaintance of yours has...

You can also just send a report what you are pleased with and what you are not so pleased with **FindExe**. Comments are welcome either!

Error- Problem-Reports:
Please document problems in a way that enables me to reproduce them easily. Otherwise I may not be able to work on (and possibly solve) the problem.

Updates / Corrections:
I'm sorry, that I cannot promise anything concerning this. At first try to get a new version the same way that you got this one. If you do not succeed, but urgently need an update, we might find another way, if you are willing to bear the cost...

Other:
My English is not perfect - but I hope good enough for you to understand. Comments on this (and corrections) are welcome.

Hope you enjoy the tool (as I do)!

Munich, 8th of August 1995, Dieter Hollmann

Windows' Search Rules

Order of the Search for executable Files:

search in ...

1 the current directory.
2 the Windows directory.
3 the Windows system directory.
4 the directory that contains the executable file for the current application (usually called the "program directory").
5 the directories listed in the PATH environment variable.
6 further network search directories, if a network and additional assignments are available (usually equivalent to the PATH variable).

If the file name contains a path specification, only this path is searched. Otherwise the rules mentioned above apply.

FindExe searches for files according to the same rules to simulate the behaviour of Windows.

Note:

FindExe does not accept a path specification in the file name (see also: "Find"- Edit Box). This would be in contrast to the sense of the implemented search mechanisms. Therefore the last statement (concerning a path specification within the file name) only applies to the *Windows*- mechanism. All other rules *do* apply to **FindExe** as well.

Annotation:

Some programs, such as **Windows Help** also look for data files in this manner.