Programmer's File Editor Help File

The PFE help file contains full information on the current version of the editor. There is no written documentation, but you will be able to find all that you need both to begin work with PFE, and to use its more advanced features, in here.

When you're using PFE's dialog boxes, you can get help on the exact steps you should take by clicking the **Help** button or pressing the **F1** key.

Like all Windows help files, this one contains many hypertext links which take you from one topic to another in a structured way. Also, most topics set up the **Up** button above the text display area to take you to the most appropriate menu topic, so you can easily retrace your steps from a lengthy session of following links.

If you're not sure how to use the Windows Help System, select the **How To Use Help** item from the **Help menu** at the top of this window, or from PFE's own help menu.

Introduction To PFE	This section gives you general information on PFE, tells you where you can get the latest version, and contains notes specific to this release
Procedures	This section tells you how to go about doing things in PFE, and explains PFE's features
Commands	This section gives you information on each of PFE's menu commands. You can look here for a summary of what a menu command does, then follow the links into the **Procedures** section for a fuller explanation
Reference	This section contains reference information: there are lists of the default key bindings, of the various bindable functions, the format of the initialisation file and so on

Introduction To PFE

Welcome to Programmer's File Editor, a programmer-oriented text editor for Windows 3.1x, Windows for Workgroups 3.1x and Windows NT 3.10 on Intel platforms.

PFE is a standard Windows Multiple Document Interface (MDI) application, which means that you can operate it in the same way as any other standard Windows application. It features many powerful facilities, which you will find described in the sections in this help file.

This introductory section gives you an overview of what PFE is, tells you where you can the latest version, and so on

What Is Programmer's File Editor?

Programmer's File Editor (PFE) is a large-capacity text file editor for Windows 3.1, Windows For Workgroups 3.1, and Windows NT on Intel platforms. It is oriented towards those who use Windows as their program development environment, and so incorporates many features that make it a convenient work management system.

PFE's capacity is essentially limited only by the total amount of memory available on your system. There are no editor-imposed limits on the number of files that you can edit simultaneously, nor on the number of edit windows that you may have open. There is no limit on the size of file that can be handled, and none on the number of lines that a file may contain.

PFE adheres strictly to the Windows MDI conventions. You can invoke most commands and facilities from menus; you can move around with a mouse or with the standard keyboard shortcuts; you can cut and paste from the clipboard, and so on.

You can reconfigure the use of keys, so that if you don't like the built-in way PFE works, you can change which keys do what to suit your preferences. If you like, you could have commands invoked by two-character key sequences like **Esc G** and **Ctrl+K Ctrl+B**; or you could use key sequences such as **Alt+F** and **Alt+S** - almost all keyboard keys can be mapped, in any combination.

PFE is able to run DOS commands, such as compilers, and to capture their output into windows for inspection. This lets you use it as an integrated development environment, cutting down the amount of work you need to do to build and test your applications. You can also quickly launch the application you're developing, and you can configure details of the Windows tools that you use, so that these too can be launched with only a few mouse clicks.

There are also some features that let you build files more easily. You can define sets of templates - standard lines of text - that you can insert into the file you're editing with just a few mouse clicks. You can group the templates you work with into distinct files, and load them for use automatically.

Why should you use PFE? If you already have a favourite editor, fine. PFE is not sold for money, so the author has no need to persuade you to do something you don't want. However, you might like it; and you are also very welcome to contribute ideas for improvements. There is a great deal of scope for changing and improving things, so let the author know what you think.

Disclaimer

Programmer's File Editor is supplied on an *as-is* basis. The author offers no warranty of its fitness for any purpose whatsoever, and accepts no liability whatsoever for any loss or damage incurred by its use.

PFE is not a supported product. The author accepts no commitment or liability to address any problems that may be encountered in using it; however, PFE is continually being developed and improved, so he is always interested to hear about any bugs or deficiencies.

Where To Get It

Programmer's File Editor is available from a wide range of archive sites throughout the world. New releases are uploaded by the author to three major archives:

- For those at UK Higher Education establishments on the JANET network, copies are placed on the **micros.hensa.ac.uk** archive

- For those with Internet access, copies are posted to **ftp.cica.indiana.edu** and **wsmr-simtel20.army.mil**

The versions on these archives will be the definitive latest releases. Note that files can take a long time to be placed in their final directories on CICA, so check in the uploads area (and any subdirectories this contains) first.

The author also passes releases to the moderator of the **comp.binaries.ms-windows** group on Usenet, and posts announcements to the **comp.os.ms-windows.announce** group.

Though not posted there by the author directly, copies will normally be available on **CompuServe** within a few weeks of release, in the **WINSHARE** forum.

From here it will gradually spread to other repositories: if you know of an archive or BBS in your part of the world that doesn't yet have the latest release, please upload a copy there too.

Using And Distributing PFE

Programmer's File Editor may be freely used for any purpose. You may distribute it to anyone, and you may place it on any archive or bulletin board system for wider access. You may not charge anyone for it other than a reasonable fee to cover your distribution costs.

Normally, you should distribute PFE in the form as supplied by the author; however, you may repackage it to suit the conventions and needs of an archive or bulletin board system if you wish.

Those wishing to distribute an unmodified executable form of PFE with commercial products are invited to contact the author.

Notes On This Release

Programmer's File Editor is still under active development. Although it is reliable in operation - many people all over the world now use it as their preferred editor, and the author does all the development work on PFE with it - not all the planned facilities have yet been added to it, and some facilities may in the future be provided in a different way.

The **changes.txt** file that accompanies each release gives you a history of the new facilities that have been added and the bugs that have been fixed, and a list of the bugs that have been reported but not yet cleared.

For help on the question asked most frequently about PFE, see the file **faq.txt** that accompanies each release.

The author welcomes feedback from users, and ideas for further facilities.

Notes Specific To The Windows NT Version

Notes Specific To The Windows NT Version

The Windows NT version of PFE for i386 platforms, distributed as **pfe32.exe**, has been built and tested on Windows NT 3.10 with the CSD001 patch set applied. It has not been tested against any Windows NT 3.5 Beta release.

The **changes.txt** file that accompanies each release lists any bugs and problems specific to the Windows NT version; items are marked **[PFE32]**.

Both the Windows/16 and Windows NT versions use the same format of initialisation file and key mapping files. However, since normally both systems are installed in the same Windows directory, the file names will differ between the two versions.

The Windows NT version has not been tested with the initialisation file mapped to the registry.

The format of template files is also common to both versions.

The Windows/16 and Windows NT versions run independently of each other. An attempt to start the Windows NT version will never instead pass control to a Windows/16 version running under the Win16 subsystem, and vice versa.

There are currently no plans to produce binary versions of PFE32 for MIPS or DEC Alpha systems.

Acknowledgements

I'm happy to be able gratefully to acknowledge here the help of those who have contributed ideas and testing time towards this project. Bug reports, comments and suggestions are *always* welcome from *everyone*. I'll try to respond to all e-mail messages; time just may not permit postal replies, I'm afraid, but everything will be read and considered.

Icons and bitmaps were designed by Dave Bleasdale (copyright assigned)

Those who have worked with the beta releases, and found the time to pass on their findings, ideas and suggestions include (and if I've mislaid *your* name in my ever-less-organised pile of mail printouts and haven't given you the credit and thanks you should have, please accept my apologies, and do let me know!):

Peter Arien, Mark Atkinson, Dave Bleasdale, Vincent Broman, Jeremy Brown, Paul Butcher, Adrian Crisan, Richard Crossley, Sal Denaro, Kent Dietz, Eyal Doron, Charles Drake, Danek Duvall, David Elliott, Dave Ellis, Andy Errington, Ken Ewart, Roger Foss, Andreas Furrer, Armand Gaudette, Bob Glauz, Marc Goldman, Jean-Luc Guimpier, Mark Hadfield, Roger Hadgraft, Bruce Hamilton, Doug Harber, Harry Hedgehog, Andreas Hestermeyer, Paul Holmes-Higgin, Don Howes, Frank van der Hulst, Dave Ingles, Andy Jacobs, Omer van der Horst Jansen, Jim Kape, Michael Krieg, Martin Kalugin, Cheong Koo, Santanu Lahiri, Michael John Marshall, Jim Martin, Brian McCashin, Rolf Michelsen, David M. Miller, Pentti Moilanen, Ron Murray, Conor Nolan, David Nugent, Tapio Peltola, Leonid Pryadko, Dave Postill, Gareth Rees, Jeff Rife, Steve Ronsen, Robert Schoolfield, Jim Sewell, Dan Shearer, Jeffrey Siegal, Nathan Silva, Gavin Silver, David Sowa, Richard Spence, Bob Stanojevich, Mike Strock, Julian Templeman, Russell Thamm, Alan Walford, Jack J. Woehr, Orinoco Womble

Without their encouragement, bug detection and splendid notions PFE would be much inferior to what it now is. Thanks, guys.

PFE Menu Commands

The menu bar shown at the top of the main window contains a selection of items that you can use to perform many of PFE's operations. Not all of the menu items on the menu bar, and not all of those on each of the pull down menus, will appear if you don't have a file open - for example, there'll be no **Edit** item on the menu bar if there's no file to edit.

The File Menu	File-related commands that let you open, close and save files, etc.
The Edit Menu	Text-altering commands that let you search for strings and replace them, use the clipboard, etc.
The Options Menu	Commands for setting the screen font, manipulating file and window modes, controlling the status bar and tool bar, etc.
The Template Menu	Commands for handling all aspects of templates
The Execute Menu	Commands for running DOS commands and Windows tools
The Macro Menu	Commands for using keyboard macros
The Window Menu	Commands for arranging, duplicating and selecting windows
The Help Menu	Commands that start the Windows help engine

The File Menu

This item on the menu bar contains file-related options. Not all of the items listed here will appear on the menu if you don't have any files open.

File New
File Open
File View
File Insert
File Save
File Save As
File Save All
File Write
File Name
File Mail
File Close
File Close All
File Print
File Print Setup
File Exit

Most-Recently-Used List

File New

This menu item creates a new, empty edit window in which you can type text.

Default command key: **Ctrl+N**

Tool bar action: 

Desktop mouse action: **Ctrl + Click Right Button**

The window will have no file name associated with it. The tab size and other settings will be the values you set up for "new files" with the **Options Default File/Window Modes** command

About Creating a New File

File Open

Opens one or more existing files for editing.

Default command key: **Ctrl+O**

Tool bar action: 

Desktop mouse action: **Click Right Button**

Selecting this item shows you a standard dialog box, from which you can specify the file or files you want to work on.

About Opening Existing Files

File View

Opens one or more existing files in *read only* mode. You will be able to examine their contents, but not alter them.

Default command key: None

Tool bar action: **Shift +** 

Desktop mouse action: **Shift + Click Right Button**

Selecting this item starts a dialog that is similar to the one produced by the **File Open** command. However, this dialog does not have a **Read Only** check box.

You can remove *read only* mode from a file with the status bar or with the **Options Current File/Window Modes** command.

About Opening Existing Files

File Insert

Inserts an existing file anywhere into the text you're editing.

Default command key: None

Selecting this item shows you a standard dialog box, from which you can specify the file to insert.

About Inserting An Existing File

File Save

Saves the contents of a file you're editing to a disk file.

Default command key: **Ctrl+S**

Tool bar action: 

If you haven't associated a name with the file yet, PFE treats this menu item as if you'd selected the **File Save As** command instead.

You cannot use this option to save a template that you're editing into a template file. Instead, use the **Template Store** or **Template Store As** commands.

If the **File Modes** of the current file so specify, and if a file of the same name already exists, PFE will maintain a backup copy, allowing you quickly to revert to a previous version of the file.

About Saving A File To Disk

File Save As

Saves the contents of a file to disk, allowing you to specify the file name.

Default command key: None

Tool bar action: 

Selecting this item shows you a standard dialog box, from which you can specify the name of the disk file you want to save the data to.

Using this option changes the name associated with the window, so that future **File Save** commands will write to the same disk file automatically.

The dialog allows you to specify that, if you save to a file of the same name as one that already exists, a backup copy is kept, allowing you quickly to revert to a previous version of the file.

About Saving A File To Disk

File Save All

Saves all altered files, stores all altered templates into template files, and saves all altered template files in a single operation.

Default command key: None

PFE will check through all the files and templates that you currently have open, saving them if they have altered.

The process is done in several stages. First, all altered named files are written to disk; then all altered templates that are already associated with template files are stored in the in-memory copy of the relevant template files.

Next, PFE will ask you if you want to save each altered file that does not have a name. You can elect not to save a file if you wish; if you do want to save it, you'll see a dialog asking you to give a file name.

After this, PFE will ask you if you want to store each unnamed template. If you elect to store one, you'll be asked to supply a name for the template, and to specify the template file it is to be stored in. If you don't have any template files attached, you'll see a warning message, and this step will be skipped.

Finally, PFE will save all the attached template files that have been altered to disk.

Clicking **Cancel** in any of the dialogs that occur in this process will cancel the entire operation.

About Saving A File To Disk

File Write

Writes the contents of a file to disk, allowing you to specify the file name

Default command key: None

Selecting this item shows you a standard dialog box, from which you can specify the name of the disk file you want to write the data to.

The dialog allows you to specify that, if you write to a file of the same name as one that already exists, a backup copy is to be kept, allowing you quickly to revert to a previous version of the file.

Unlike the **File Save As** command, this command does *not* permanently associate the name of the disk file with the window showing the data.

About Saving A File To Disk

File Name

Changes the disk file name associated with a window.

Default command key: None

Selecting this item shows you a standard dialog box, from which you can specify the name of the disk file you want to associate with the window.

After using this command, the **File Save** command will write the data to disk in a file with the name you've specified. The file is marked as having been altered, so that the **File Save** command will be immediately available for you to save the file with its new identity

This command does not affect any existing disk file.

About Saving A File To Disk

File Mail

Mails the data in the current window to someone else, using a MAPI-compliant mailer.

Default command key: None

The command saves the data in the current window to a temporary file, then calls the mail system to prompt you for delivery instructions. The file will be mailed as an attachment.

Since the mailer may itself need to make a further copy of the file, you should ensure that you have adequate disk space available for this operation.

You will not be able to select this option if PFE does not detect a MAPI-compliant mailer in your system configuration when you start it.

About Mailing A File

File Close

Closes the file being shown in the current window. The current window, and all other windows showing the same file, will be closed.

Default command key: None

If you have changed the file concerned, and haven't saved the changes to disk, PFE will prompt you and allow you to save the changes, to discard them, or to cancel the close.

About Closing A File

File Close All

Closes all the files that you're working on. All windows will be deleted.

Default command key: None

If you have changed any of the files concerned, and haven't saved the changes to disk, PFE will prompt you and allow you to save the changes, to discard them, or to cancel the entire close operation.

About Closing A File

File Print

Prints all or part of the current file.

Default command key: **Ctrl+P**

Tool bar action: 

The command starts a dialog that allows you to specify the details of how the file is to be printed and on which printer.

By default, PFE uses the same printer that you specified the last time you printed a file; default details of the printing, such as line folding and page headers, are taken from the window modes set on the current window.

You can also change the printer setup and the font to be used.

About Printing A File

File Print Setup

Selects the printer to be used by default for printing files, alters its setup details, and selects the font to be used.

Default command key: None

Tool bar action: **Shift +** 

The command starts a dialog that allows you to configure the various details. The settings are recorded, and will be used the next time you start PFE.

About Printing A File

File Exit

Terminates your PFE session

Default command key: None

This command will end your PFE session. If any of the files you're working with have changed, and you haven't saved the changes to disk, PFE will prompt you for each one in turn. You can choose to save the changes, discard them, or cancel the closedown.

The **Exit Windows** item in PFE's system menu allows you to end your PFE session *and* terminate Windows itself in a variety of ways.

About Ending a PFE Session

File Most-Recently-Used List

This is not a single menu item, but a list that PFE adds automatically to the end of the **File** menu.

Whenever you open a file, PFE records the name in the list, ordering it so that the files you have used most recently always appear at the top. You can then re-open any file in the list simply by clicking on the menu item.

Unless you started PFE in multi-instance mode with a command line option or a setting in the initialisation file, PFE records the list in its initialisation file whenever you exit it. It will always re-instate the list when you start the next session.

You can control the size and display details of the list by editing the **[options]** section of the initialisation file:

- The **mru-list-size** value sets the maximum number of file names that PFE records in the list: you can set this at any value between 0 and 64. PFE will take longer to initialise itself, and to write information to the initialisation file on termination, the higher the value used.

- The **mru-files-shown** value controls how many names are shown on the file menu itself: you can set this to any value between 0 and 8. On standard VGA screens, values above 5 may make the **File** menu too long to be conveniently shown on the screen.

When the list contains more files than can be displayed on the **File** menu, an extra menu command **More Files** is automatically added (if you've set the **mru-files-shown** value to be zero, the menu command will be **Recent Files**). The command starts a dialog that shows you the entire contents of the list, and lets you pick the file you want to open.

The Edit Menu

This item on the menu bar contains options concerned with altering text. It will not be shown if you have no files open.

Edit Undo
Edit Clear Undo
Edit Cut
Edit Copy
Edit Paste
Edit Select All
Edit Select Word
Edit Delete Line
Edit Delete To End Of Line
Edit Show Caret
Edit Goto Line
Edit Find
Edit Replace
Edit Repeat Last Find
Edit Repeat Last Replace
Edit Text

Edit Undo

Undoes the last edit action, returning the current file to its previous state

Default command key: **Ctrl+Z**

Tool bar action: 

The text of this command on the **Edit** menu is not constant, but changes to indicate the nature of the action that will be undone if you use it. For example, if the last edit action you performed was to cut an area of text to the clipboard, the menu command will be **Edit Undo Cut**. Similarly, **Edit Undo Drag-Drop** indicates that the last edit action you performed was a drag-and-drop move or copy of text, and so on.

When there is no edit action to undo, or PFE is unable for some reason to reverse the last change you made, the menu command will show **Can't Undo**, and you will not be able to select the command.

Normally PFE records the last 32 edit actions that you have performed on each file that you are editing. You can change this to record more or fewer actions by setting various options in the **[options]** section of the initialisation file.

About Undoing Edit Actions

Edit Clear Undo

Clears the recorded details of all the edit actions performed up to this point for the current file, returning the memory used to the system

Default command key: None

This command allows you to free the memory that PFE is using to record the changes you have made to the current file. Depending on what edit changes you have made to the current file, a significant amount of memory could be freed for other uses.

Once you have used this command, you will be unable to undo any preceding edit changes to the current file.

The command will ask you to confirm that you wish to take this action before carrying it out.

Normally PFE will automatically perform this action when you write a file to disk. You can change this behaviour, so that undo actions are discarded only when you explicitly command it, by setting the **save-clears-undo** value in the **[options]** section of the initialisation file.

About Undoing Edit Actions

Edit Cut

Deletes the highlighted text in the current window, placing it on the clipboard.

Default command key: **Ctrl+X**

Tool bar action: 

After this operation, you will be able to paste the deleted text into another PFE window, or into another application

About Cutting, Copying and Pasting Text

Edit Copy

Copies the highlighted text in the current window onto the clipboard.

Default command key: **Ctrl+C**

Tool bar action: 

After this operation, you will be able to paste the deleted text into another PFE window, or into another application.

Normally the text you've copied remains highlighted after this operation. You can arrange for the highlight to be removed by including the line

```
deselect-on-copy=1
```

in the **[options]** section of the initialisation file.

About Cutting, Copying and Pasting Text

Edit Paste

Pastes the contents of the clipboard into the current window at the position of the caret

Default command key: **Ctrl+V**

Tool bar action: 

You will only be able to perform this operation if the clipboard contains data in text format. PFE does not support pasting of bitmaps or other clipboard formats.

About Cutting, Copying and Pasting Text

Edit Select All

Highlights the entire contents of the current window.

Default command key: None

You can perform this operation wherever the caret currently is in the file. After it the caret will be positioned at the end of the file.

About Selecting Text

Edit Select Word

Highlights the word that the caret is currently in.

Default command key: None
Mouse shortcut: Double-click the left button

About Selecting Text

Edit Delete Line

Deletes the entire line containing the caret.

Default command key: **Ctrl+Shift+K**

This command deletes the entire line, wherever the caret is within it. The caret moves to the start of the following line.

About Deleting Text

Edit Delete to End of Line

Deletes the text from the position of the caret to the end of the line containing it

Default command key: **Ctrl+K**

If the caret is at the start of a line that is completely empty, the line following is moved up to close the gap.

About Deleting Text

Edit Show Caret

Moves the data being shown in the current window so that the caret is visible, and the line containing it is as close as possible to the middle of the window

Default command key: **Ctrl+F5**

Edit Goto Line

Moves the caret to the start of a specific line.

Default command key: **Ctrl+G**
Used from status bar?: **Yes**

The command starts a dialog that allows you to specify the line number of the target line. You can also choose to highlight all the text between the current position of the caret and the target line.

You can move to either an absolute line number, or move up or down by a number of lines from the current position.

About Moving to Specific Lines

Edit Find

Searches the current window for a text string.

Default command key: **F2**

Tool bar action: 

The command starts a standard dialog that lets you specify the details of the search. The dialog remains visible, and can be re-used for another search in the same window, or in another window.

Any highlighted text in the current window will be set up as the default string to be searched for.

About Finding Text

Edit Repeat Last Find

Repeats the last search operation, without prompting for details.

Default command key: **Shift+F2**

Tool bar action: **Shift +** 

The command repeats the last search operation, using the details last set with the **Edit Find** command.

If you have not previously performed a search operation, the command acts as the **Edit Find** command and shows the standard dialog.

If the **Edit Find** dialog is visible, the command activates it rather than performing a search.

About Finding Text

Edit Replace

Searches the current window for a text string, and optionally replaces occurrences with another string.

Default command key: **F3**

Tool bar action: 

The command starts a standard dialog that lets you specify the details of the operation. The dialog remains visible, and can be re-used for another search in the same window, or in another window.

Any highlighted text in the current window will be set up as the default string to be searched for.

Using the dialog, you can either scan through the file, deciding whether or not to replace each occurrence of a match; or you can elect to replace all occurrences automatically.

About Replacing Text

Edit Repeat Last Replace

Repeats the last replace operation, without prompting for details.

Default command key: **Shift+F3**

Tool bar action: **Shift +** 

The command repeats the last replace operation, using the details last set with the **Edit Replace** command.

If you have not previously performed a replace operation, the command acts as the **Edit Replace** command and shows the standard dialog.

If the **Edit Replace** dialog is visible, the command activates it rather than performing a replacement.

About Replacing Text

Edit Text

Shows a popup sub-menu containing text-related commands.

Edit Text Transpose Characters
Edit Text Uppercase Selection
Edit Text Lowercase Selection
Edit Text Indent
Edit Text Undent
Edit Text Insert ASCII Code
Edit Text Match Brace
Edit Text Match Brace Select
Edit Text Widen Brace Select

Edit Text Transpose Characters

Transposes the character to the right of the caret and the character to the left of the caret.

Default command key: None

It has no effect if the caret is positioned at the start or the end of a line.

About Text Processing

Edit Text Uppercase Selection

Changes all lower-case characters in the currently highlighted text to upper-case characters.

Default command key: None

About Text Processing

Edit Text Lowercase Selection

Changes all upper-case characters in the currently highlighted text to lower-case characters

Default command key: None

About Text Processing

Edit Text Indent

Moves the text in one or more lines right by one tab stop.

Default command key: None

If you do not have any text highlighted, PFE indents only the line that the caret is in. Otherwise, it indents all the lines containing highlighted text.

About Text Processing

Edit Text Undent

Moves the text in lines left by one tab stop.

Default command key: None

If you do not have any text highlighted, PFE undents only the line that the caret is in. Otherwise, it undents all the lines containing highlighted text.

Lines that do not start with white space are not affected.

About Text Processing

Edit Text Insert ASCII Code

Inserts a character specified by its ASCII code

Default command key: None

The command starts a dialog that lets you specify the ASCII code of the character to insert. You can choose from a preset list of control characters (ASCII codes 1 to 31) or type the actual decimal code value yourself.

About Text Processing

Edit Text Match Brace

Moves the caret to a matching brace character

Default command key: **Ctrl+B**

The action of this command depends on the language type defined in the current window's window modes.

Language type "(none)" or "TeX"

The command checks that the caret is positioned on a character that is either an opening brace from the set **{ [(** or **<**, or is a closing brace from the set **}])** or **>**.

If so, PFE scans the window in the appropriate direction to locate the brace that matches it, having regard to nested brace pairs. If a match is found, the caret is moved to lie on the caret; if no match is found, the caret remains where it is.

In scanning for the matching brace character no regard is paid to any syntactic elements of any kind in the text.

Language type "C"

The command checks that the caret is positioned on a character that is either an opening brace from the set **{ [** or **(**, or is a closing brace from the set **}]** or **)**.

If so, PFE scans the window in the appropriate direction to locate the brace that matches it, having regard to nested brace pairs. If a match is found, the caret is moved to lie on the caret; if no match is found, the caret remains where it is.

In scanning for the matching brace character regard is paid to some elements of C language syntax. Characters within comments, or inside pre-processor directives, are not scanned; nor are characters defined as character literals. However, depending on where the caret initially lies in relation to syntactic elements, on some occasions it may not be possible to make an unambiguous determination of the correct matching brace. This will happen especially if the syntactic elements in the text are invalid or not yet complete.

If it is required to both move to a matching brace *and* highlight the text between the braces, you can use either the **Edit Text Match Brace Select** or **Edit Text Widen Brace Select** commands.

About Text Processing

Edit Text Match Brace Select

Moves the caret to a matching brace character, highlighting all the text between and including the brace characters

Default command key: **Ctrl+Shift+B**

The action of this command depends on the language type defined in the current window's window modes.

Language type "(none)" or "TeX"

The command checks that the caret is positioned on a character that is either an opening brace from the set **{ [(** or **<**, or is a closing brace from the set **}])** or **>**.

If so, PFE scans the window in the appropriate direction to locate the brace that matches it, having regard to nested brace pairs. If a match is found, the caret is moved to lie on the caret and all the text between and including the brace characters is highlighted; if no match is found, the caret remains where it is.

In scanning for the matching brace character no regard is paid to any syntactic elements of any kind in the text.

Language type "C"

The command checks that the caret is positioned on a character that is either an opening brace from the set **{ [** or **(**, or is a closing brace from the set **}]** or **)**.

If so, PFE scans the window in the appropriate direction to locate the brace that matches it, having regard to nested brace pairs. If a match is found, the caret is moved to lie on the caret and all the text between and including the braces is highlighted; if no match is found, the caret remains where it is.

In scanning for the matching brace character regard is paid to some elements of C language syntax. Characters within comments, or inside pre-processor directives, are not scanned; nor are characters defined as character literals. However, depending on where the caret initially lies in relation to syntactic elements, on some occasions it may not be possible to make an unambiguous determination of the correct matching brace. This will happen especially if the syntactic elements in the text are invalid or not yet complete.

If it is required to both move to a matching brace *without* highlighting the text between the braces, use the **Edit Text Match Brace Select** command.

About Text Processing

Edit Text Widen Brace Select

Highlights the next largest area of text delimited by matching brace characters and including either the character under the caret, or the current selection.

Default command key: **Ctrl+Shift+W**

The action of this command depends on the language type defined in the current window's window modes.

Language type "(none)" or "TeX"

The command attempts to locate a matching pair of brace characters that contain either the character under the caret, or the whole of any highlighted text. It regards opening braces as any character from the set **{ [(** or **<**, and closing braces as the corresponding characters from the set **}])** or **>**.

If a matching brace pair is found, the text between and including them is highlighted.

In scanning for the matching brace characters no regard is paid to any syntactic elements of any kind in the text.

Language type "C"

The command attempts to locate a matching pair of brace characters that contain either the character under the caret, or the whole of any highlighted text. It regards opening braces as any character from the set **{ [** or **(**, and closing braces as the corresponding characters from the set **}]** or **)**.

If a matching brace pair is found, the text between and including them is highlighted.

In scanning for the matching brace characters regard is paid to some elements of C language syntax. Characters within comments, or inside pre-processor directives, are not scanned; nor are characters defined as character literals. However, depending on where the caret initially lies in relation to syntactic elements, on some occasions it may not be possible to make an unambiguous determination of the correct matching brace. This will happen especially if the syntactic elements in the text are invalid or not yet complete.

Repeated use of this command will have the effect of selecting increasingly larger sections of text, enabling you easily to locate and access large delimited blocks.

About Text Processing

The Options Menu

This item on the menu bar contains various system configuration options. Not all the items will appear if you don't have a file open.

Options Current File/Window Modes
Options Default File/Window Modes
Options Reset File/Window Modes
Options Screen Font
Options Status Bar
Options Tool Bar
Options Key Mapping
Options DDE Server

Options Current File/Window Modes

Starts a dialog that allows you to configure the modes that apply to the current window, and to the file that is being shown in the current window.

Default command key: None

About File and Windows Modes

Options Default File/Window Modes

Starts a dialog that allows you to configure the modes that PFE applies to windows and files when you open existing files or create new ones.

Default command key: None

About File and Windows Modes

Options Reset File/Window Modes

Resets the window modes that apply to the current window, and the file modes that apply to the file it's showing, to the default values.

Default command key: None

If the current window is showing a file that has a file name associated with it, the modes set are those that match the file type.

If the window is showing a file that so far has no name associated, the modes set are those appropriate to newly-created files. In all other cases, the modes used are those for files with no specific file type.

About File and Windows Modes

Options Status Bar

Turns display of the status bar on or off

Default command key: None

If the menu option is checked, the status bar will be visible.

About The Status Bar

Options Screen Font

Shows a popup sub-menu that allows you to set the details of the font that PFE uses in windows.

PFE uses the same screen font in all windows. The details you set are recorded, and will automatically be uses the next time you start PFE.

ANSI
OEM
System
Other

About Screen Fonts

Options Screen Font ANSI

Specifies that text is to be shown using the system's standard ANSI font. This will commonly be a Courier typeface, but this will be system-dependent.

Default command key: None

About Screen Fonts

Options Screen Font OEM

Specifies that text is to be shown using the system's standard OEM font.

Default command key: None

About Screen Fonts

Options Screen Font System

Specifies that text is to be shown using the standard fixed pitch system font.

Default command key: None

About Screen Fonts

Options Screen Font Other

Lets you select any available fixed pitch screen font to use in all edit windows.

Default command key: None

The command starts a standard dialog that allows you to specify the typeface to be used, the type style (*i.e.* regular, bold or italic), and the type size. The selection applies to all windows.

About Screen Fonts

Options Tool Bar

Shows a popup sub-menu that contains items used to control the position and appearance of the tool bar.

PFE records the state of the tool bar, and will set it to that state automatically when you next start it.

Options Tool Bar Hide
Options Tool Bar Show
Options Tool Bar Top
Options Tool Bar Bottom
Options Tool Bar Left
Options Tool Bar Right
Options Tool Bar Floating

About The Tool Bar

Options Tool Bar Hide

Removes the tool bar from the screen.

Default command key: None

About The Tool Bar

Options Tool Bar Show

Makes the tool bar visible on the screen in its last position.

Default command key: None

About The Tool Bar

Options Tool Bar Top

Sets the tool bar to appear as a bar along the top edge of the main PFE window.

Default command key: None
Tool bar action: **Click + Drag**

About The Tool Bar

Options Tool Bar Bottom

Sets the tool bar to appear as a bar along the bottom edge of the main PFE window.

Default command key: None
Tool bar action: **Click + Drag**

About The Tool Bar

Options Tool Bar Left

Sets the tool bar to appear as a bar down the left edge of the main PFE window.

Default command key: None
Tool bar action: **Click + Drag**

About The Tool Bar

Options Tool Bar Right

Sets the tool bar to appear as a bar down the right edge of the main PFE window.

Default command key: None
Tool bar action: **Click + Drag**

About The Tool Bar

Options Tool Bar Floating

Sets the tool bar to appear as a floating window that you can position anywhere within the main PFE window.

Default command key: None
Tool bar action: **Click + Drag**

A floating tool bar window always remains visible above any edit windows.

About The Tool Bar

Options Key Mapping

Maps keys to functions.

Default command key: None

The command starts a dialog that enables you to set which operations are to be performed when you press particular keys. You can save the mappings you specify so that PFE will use them automatically the next time you start it. You can also save them to as individual key mapping files, which you can load for specific purposes as you require.

By default, PFE uses a set of built-in key mappings.

About Key Mapping

Options DDE Server

Turns the DDE server facilities on and off.

Default command key: None

This command can be used at any time to start and stop PFE's DDE server system. Normally, PFE will act as a DDE server by default, but if you've configured the system to run multiple instances by default, or started PFE with a "**/m**" command line option, the DDE server will not be running.

The menu item will be checked if the DDE server is running. You should normally try to have only one instance of PFE that is a DDE server running at any time, as it's not predictable which one another application will connect to.

Controlling PFE Over A DDE Link

The Template Menu

This menu item contains options that relate to handling templates. Not all the options will be shown if you don't have any files open.

Template Attach File
Template Detach File
Template Create File
Template Save File
Template Insert
Template Edit
Template New
Template Store
Template Store As
Template Delete
Template Insert Mark
Template Find Mark

About Templates

Template Attach File

Attaches a template file, loading into memory so the templates within it can be used.

Default command key: None

The command starts a dialog that allows you to browse the files on your disk for template files, which have type "**.tpl**".

You can arrange for PFE automatically to attach the template files you use frequently, and also automatically to attach one template file specific to the directory you start it in.

About Templates

Template Detach File

Detaches a template file, unloading it from memory.

Default command key: None

The command starts a dialog that allows you to specify which attached template files you wish to detach. Once a template file is detached, you cannot use the templates it contains.

You can't detach a template file if you're currently editing any of the templates it contains. You will need to save any changes you want to keep with the **Template Store** or **Template Store As** commands, and then close the windows.

About Templates

Template Create File

Creates an empty template file and automatically attaches it.

Default command key: None

The command starts a dialog that allows you to choose the name of the template file to create. This should have a file type of "**.tpl**". PFE will create the file on disk and initialise it.

Template files do not contain pure text, and this is the only method by which you can create them.

About Templates

Template Save File

Saves a template file to disk after you have edited its contents

Default command key: None

The command starts a dialog that shows you the names of the attached template files that have changed. You can choose which of them to save.

Template operations such as edit, delete, new and store work only the copy of the template file in memory. Any changes you make are not recorded permanently until you use this command.

About Templates

Template Insert

Inserts a template from an attached template file into a window.

Default command key: **F9**

Tool bar action: 

The command starts a dialog that shows you the names of the attached template files, and the templates they contain. You can select any template from any file, then click **OK** to action the insert.

The template text is inserted into the current window at the position of the caret. Unlike the case of inserting a file from disk, the caret is left positioned at the *start* of the inserted text: this allows you to use the **Template Find Mark** command to fill in the positions marked as needing text input.

About Templates

Template Edit

Edits a template contained in an attached template file.

Default command key: None

Tool bar action: **Ctrl +** 

The command starts a dialog that shows you the names of the attached template files, and the templates they contain. You can select any template from any file, then press **OK** to edit it.

PFE will create a new window, and copy the text of the template into it. You can edit the template as you wish; then, when you've made the changes, you can use the **Template Store** command to update the in-memory copy of the template file.

Note that any changes you make will not be recorded permanently in the disk copy of the template file until you use the **Template Save File** command.

About Templates

Template New

Creates a new, empty window in which you can type the text of a new template.

Default command key: None

Tool bar action: **Shift +** 

The window created is not associated with either a template name, or a template file. To record what you've typed, you will need to use the **Template Store As** command to name the template and place it in a specific template file.

About Templates

Template Store

Stores a template that you've been editing into the in-memory copy of the template file that it is associated with.

Default command key: None

The previous version of the template that the template file contains will be overwritten by this command. You can store a maximum of 64 templates in a template file, and the total space used by all templates in the file cannot exceed a total of approximately 62 kilobytes.

Note that the template file on disk is not altered by this command. To save your changes permanently, you next need to use the **Template Save File** command.

If you created the window you're editing the template in with the **Template New** command, and haven't yet stored the template in a template file, this command will act like the **Template Store As** command.

About Templates

Template Store As

Stores a template that you've been editing into the in-memory copy of a template file, allowing you to specify both the template file, and the name of the template.

Default command key: None

The command starts a dialog that lets you specify the details of the operation. You should first select which of the attached template files you want to store the template file in, then specify the template name. The total space used by all templates in the file cannot exceed a total of approximately 62 kilobytes.

You can select the name of an existing template if you wish; as this will overwrite the previous contents, PFE will ask you to confirm that you really want to do this.

Note that the template file on disk is not altered by this command. To save your changes permanently, you next need to use the **Template Save File** command.

About Templates

Template Delete

Deletes templates from the in-memory copy of any attached template file

Default command key: None

The command starts a dialog that lets you specify the details of the operation. You should first select which of the attached template files you want to delete templates from, then select one or more template names from the list of those it contains.

Note that the template file on disk is not altered by this command. To save your changes permanently, you next need to use the **Template Save File** command.

About Templates

Template Insert Mark

Inserts a *template mark* into a template that you're editing

Default command key: **F6**

A template mark is a sequence of characters ("**<???>**") that you can use to mark positions in a template where you need to put variable details after you've inserted it.

When you insert a template into a file, you can use the **Template Find Mark** command to quickly move to all the template marks in turn and replace them with whatever is appropriate.

About Templates

Template Find Mark

Moves the caret to the next *template mark* in a file or a template and highlights it

Default command key: **F4**

A template mark is a sequence of characters ("**<???>**") that you can use to mark positions in a template where you need to put variable details after you've inserted it.

When you insert a template into a file, you can use this command to quickly move to all the template marks in turn and replace them with whatever is appropriate.

About Templates

The Execute Menu

This menu contains items that relate to running DOS commands and starting Windows applications.

Execute DOS Command to Window
Execute Repeat DOS Command to Window
Execute DOS Prompt
Execute Launch Application
Execute Launch Windows Tool
Execute Configure Windows Tools
Execute Control Panel
Execute File Manager
Execute Print Manager
Execute Program Manager
Execute Task Manager

About Running DOS Commands
About Launching Windows Applications
About Windows Tools
About Keyboard Macros

Execute DOS Command To Window

Executes a DOS application, such as a compiler or a built in command, and captures the output in a window.

Default command key: **F11**

Tool bar action: 

The command starts a dialog that lets you give the command line to be executed. It can be any DOS command, including built-in ones like "**dir**". You can also automatically substitute parts of the name of the file showing in the current window into the command line.

You can specify the working directory that the command is to run in. This affects only the DOS command - PFE itself will still use its previous working directory.

When you start the dialog, PFE sets the command string and the proposed working directory name to the values you set the last time you used it in this or a previous session.

You may execute only one DOS command at a time. Until the command completes, PFE will reject attempts to start another.

This command is not available if you are running the Windows/16 version of PFE in the Win16 subsystem of Windows NT, or if you are running the Windows NT version of PFE on Windows 3.1 with Win32s.

About Running DOS Commands

Execute Repeat DOS Command To Window

Repeats the last DOS command you ran with the **Execute DOS Command To Window** command, capturing the command's output.

Default command key: None

Tool bar action: **Shift +** 

PFE will repeat the last command exactly, and will show you the command output in a window when it completes. If you have not previously run a command, PFE will act as if you selected the **Execute DOS Command To Window** command and will show you the dialog.

If the command line you specified in the dialog contained substitution points requesting that PFE automatically include parts of the file name of the file showing in the current window, these will be re-evaluated before the command is executed.

This command is not available if you are running the Windows/16 version of PFE in the Win16 subsystem of Windows NT, or if you are running the Windows NT version of PFE on Windows 3.1 with Win32s.

About Running DOS Commands

Execute DOS Prompt

Starts a separate task running the DOS command processor

Default command key: **Ctrl+F11**

Tool bar action: 

PFE starts a new command processor task, which runs completely independently from it.

If you have defined a command processor with a **comspec** statement in the **[options]** section of the initialisation file, this program will be run.

If not, PFE will look at the environment variable **COMSPEC**, and if this is defined, will run the program named in it.

If neither of the above options specifies a command processor, PFE will start **command.com** (for the Windows/16 version) or **cmd.exe** (for the Windows NT version)

About Running DOS Commands

Execute Launch Application

Launches a DOS or Windows application as a task running independently of PFE

Default command key: **Shift+F11**

Tool bar action: 

The command starts a dialog that lets you give the command line to be executed. It can specify any DOS or Windows application.

You can specify the working directory that the application is to run in. This affects only the application - PFE itself will still use its previous working directory.

When you start the dialog, PFE sets the command string and the proposed working directory name to the values you set the last time you used it in this or a previous session.

Applications launched by this command run totally independently of PFE, and their output is *not* captured. To run something like a compiler, whose output you want to catch in an edit window, use the **Execute DOS Command To Window** command.

About Launching Windows Applications

Execute Launch Windows Tool

Launches one of the Windows applications configured in the list of Windows Tools

Default command key: **F12**

Tool bar action: **Shift +** 

This command gives a quick way of starting a pre-defined Windows application such as the Dialog Editor or a debugger.

When you select it, PFE shows you a dialog that gives a list of the configured Windows tools. You can select one from the list, modify the associated command line and working directory if you wish, and press **OK** to launch it.

Applications launched in this way run independently of PFE, and their output is *not* captured. To run something like a compiler, whose output you want to catch in an edit window, use the **Execute DOS Command To Window** command.

To set up or edit the list of available Windows tools, use the **Execute Configure Windows Tools** command.

About Windows Tools

Execute Configure Windows Tools

Configures details of the defined Windows Tools.

Default command key: **Shift+F12**

Tool bar action: **Ctrl +** 

This command starts a dialog that lets you add or delete entries from the list of Windows tools, or modify existing ones.

About Windows Tools

Execute Control Panel

Starts the Windows Control Panel application.

Normally PFE will execute the standard Control Panel from **control.exe**. However, you can specify the command line to be executed with a **control-panel** entry in the **[managers]** section of the initialisation file.

Execute File Manager

Starts the Windows File Manager application.

Normally PFE will execute the standard File Manager from **winfile.exe**. However, you can specify the command line to be executed with a **file-manager** entry in the **[managers]** section of the initialisation file.

Execute Print Manager

Starts the Windows Print Manager application.

Normally PFE will execute the standard Print Manager from **printman.exe**. However, you can specify the command line to be executed with a **print-manager** entry in the **[managers]** section of the initialisation file.

Execute Program Manager

Starts the Windows Program Manager application.

Normally PFE will execute the standard Program Manager from **progman.exe**. However, you can specify the command line to be executed with a **program-manager** entry in the **[managers]** section of the initialisation file.

Execute Task Manager

Starts the Windows Task Manager application.

Normally PFE will execute the standard Task Manager from **taskman.exe**. However, you can specify the command line to be executed with a **task-manager** entry in the **[managers]** section of the initialisation file.

The Macro Menu

This menu item contains commands relevant to handling keyboard macros, which allow you to record sequences of keystrokes and menu commands and replay them.

Macro Start Recorder
Macro Stop Recorder
Macro Replay

Macro Start Recorder

Starts recording your key presses and menu selections as a keyboard macro, which you can then replay

Default command key: **Shift+F7**

Tool bar action: 

Once you've selected this command, PFE records all the keys you press, and all the menu selections you make, in its keyboard macro buffer. When you've performed all the actions you want recorded, use the **Macro Stop Recorder** command to turn recording off.

You can record a maximum of 1024 key strokes or menu actions. Two-character command sequences count as a single stroke.

Things that you type in dialog boxes are not recorded, nor are mouse actions.

Once a keyboard macro has been recorded, you can replay it as often as you wish with the **Macro Replay** command.

In this version of PFE, any previously stored keyboard macro is deleted when you turn recording on. Also, you cannot save the actions you record permanently.

About Keyboard Macros

Macro Stop Recorder

Stops recording your key presses and menu selections as a keyboard macro

Default command key: **Ctrl+F7**

Tool bar action: 

Once you've selected this command, PFE no longer records the keys you press, and the menu selections you make, in its keyboard macro buffer.

Once a keyboard macro has been recorded, you can replay it as often as you wish with the **Macro Replay** command.

About Keyboard Macros

Macro Replay

Replays the current keyboard macro

Default command key: **F7**

Tool bar action: **Shift +** 

The command causes PFE to replay all the key presses and menu selections that you've recorded. If any operation causes an error - for example, you might replay a **File Save** command when no file is open - the replay is terminated.

PFE does not record what you type in dialog boxes; thus, if any recorded key starts a dialog, you will need to fill in the details whenever you replay the macro. Pressing the **Cancel** button in any dialog will cause PFE to end the macro replay.

About Keyboard Macros

The Window Menu

This menu item contains commands relevant to handling edit windows. It does not appear if you have no files open.

Window Tile Horizontal
Window Tile Vertical
Window Cascade
Window Iconize All
Window Arrange Icons
Window Select
Window Next
Window Duplicate
Window Close
Window Widen

The Window List
About Window Manipulation

Window Tile Horizontal

Arranges all the non-iconic edit windows in a tile pattern, placing them side by side so that each is as deep as possible.

Default command key: **Shift+F4**

About Window Manipulation

Window Tile Vertical

Arranges all the non-iconic edit windows in a tile pattern, placing them one above the other so that each is a wide as possible.

Default command key: None

About Window Manipulation

Window Cascade

Arranges all non-iconic edit windows in a cascade pattern

Default command key: **Shift+F5**

About Window Manipulation

Window Iconize All

Makes all edit windows into icons

Default command key: None

About Window Manipulation

Window Arrange Icons

Arranges all iconized edit windows so that their icons are visible, in rows at the bottom of the PFE desktop.

Default command key: None

Windows that are not icons are not affected by this command

About Window Manipulation

Window Select

Allows you to choose between many open windows with a selection dialog

Default command key: **Ctrl+W**

This command starts a dialog that lists the window captions of all the edit windows. You can refine the list to show only windows showing files that have associated names, windows showing un-named files, windows showing templates, or windows showing command output. You can also choose to see only windows that have been altered.

You can select the window you want to see from the dialog, and press the **OK** button to switch to it.

Window Next

Moves the focus to the next edit window that is not an icon.

Default command key: None

This command will not activate any window that is currently iconic. Windows are activated in the order in which PFE maintains their control information, which will not be the same as the ordering used by Windows itself.

About Window Manipulation

Window Duplicate

Makes an exact copy of the current window

Default command key: None

The duplicate window created by this command shows the same file, with the caret in exactly the same place. All aspects of the new window are identical to the original.

You can produce as many duplicates of a window as you wish, and edit the same file in each of them,

About Window Manipulation

Window Close

Closes the current window

Default command key: None

If the window that you close is the only one showing that particular file, this command acts as if you'd used the **File Close** command. PFE will check whether the file has been changed, and will give you the opportunity of saving them.

About Window Manipulation

Window Widen

Alters the size and horizontal position of the current window to make it as wide as possible within the PFE desktop

Default command key: None

The window's vertical size and position are not affected

About Window Manipulation

Window List

This is not a single menu item, but a list that is automatically added to the end of the **Window** menu whenever you open a window.

The list shows the window captions for up to 9 of the windows that you have open at the moment. You can switch to one of them simply by clicking the appropriate menu item.

If you have more than 9 windows open, a further command **More Windows** is added to the end of the list. This command is identical in effect to the **Window Select** command; it starts a dialog that lets you choose the window to activate.

The Help Menu

This menu lets you obtain help on using PFE, and also lets you run the Windows help engine on up to five help files of your own choice.

If you wish, you can always use the **Help Contents** command to enter the help system, and navigate through the information to find what you want. To help you locate things quickly, though, various other commands allow you to enter the help system at specific points; for example, the **Help Commands** command is useful if you want to look up a menu command to see what it does.

You can also obtain context-specific help when you're working in dialog boxes by either clicking on their **Help** buttons, or pressing the **F1** key. Many message boxes that ask you questions will also show you help if you press the **F1** key.

You can obtain interactive help on menu items and screen areas such as the tool bar and status bar by pressing **Shift+F1** or using the **Help Screen/Menu Help** command. This will let you point the mouse cursor at the menu item or screen area you want help on, and click the left mouse button to enter the help system.

The **Help Context Help** command allows you to enter a specific Windows help file using the text highlighted in the current window as a search key. This is a convenient way of obtaining help on, for example, Windows API routines.

Help Contents
Help Search For Help On
Help Screen/Menu Help
Help How To Use Help
Help Commands
Help Procedures
Help Reference
Help Context Help
Help About

User-Configurable Items

Help Contents

Starts the Windows help engine to give help on PFE. The help file is opened at the Contents page

Default command key: **F1**

Help Search For Help On

Starts a dialog that allows you to search the PFE help file for a given keyword

Default command key: None

You can type the keyword you want to search for, or you can select one of the ones shown in the list, then click the **Show Topics** button to see all the help topics that refer to that keyword.

You can then select the topic that interests you, and open the help file to show it by clicking the **Goto Topic** button

Help Screen/Menu Help

Enters the interactive help system that lets you point the mouse cursor at the menu item or screen area you want help on

Default command key: **Shift+F1**

When you use this command, the mouse cursor will change to become an arrow-and-question mark, to show that interactive help is active.

For help on an area of the status bar or on a tool bar button, point the mouse cursor at the appropriate part of the screen and click the left button once. For help on a menu item, select the item as you would do actually to use the command, and click the left button.

PFE will start the Windows help engine, and will show the topic appropriate to where you clicked the mouse button.

If you decide you don't want interactive help after using the command, you can return the mouse cursor to normal by pressing the **Escape** key.

When interactive help is operating, any keys you press will be ignored, and the tool bar, status bar and menu items will not have their normal effect.

Help How To Use Help

Starts the Windows help engine to give you help on how the help system works

Default command key: None

Help Commands

Starts the Windows help engine to give help on PFE's menu commands.

Default command key: None

Help Procedures

Starts the Windows help engine to give help on how you perform various tasks using PFE.

Default command key: None

Help Reference

Starts the Windows help engine to give help on PFE. The help file is opened at the Reference section, which gives you such items as the list of functions that you can bind to keys, and so on

Default command key: None

Help Context Help

Starts the Windows help engine to give help associated with text in the current window

Default command key: **Ctrl+F1**
Mouse shortcut: Double-click the right button

If you have some highlighted text in the current window, PFE will pass the first 64 characters of it (with leading and trailing spaces removed, and non-printable characters converted to spaces) to the Windows help engine as a *partial key*.

With no text highlighted, PFE first performs an **Edit Select Word** command to select the entire word containing the caret, then passes this to the help engine.

For preference, PFE uses the help file named by the **context-help-file** key in the **[options]** section of the initialisation file. If you have not defined this key, it will use the *first* help file named in the **[help-files]** section (these are the help files that appear as additions to the standard **Help Menu**).

If the help file contains an exact match for the text, you will see the appropriate help entry; if not, you will see a list of possible topics that might match.

Help About

Starts a dialog that gives you information about the particular version of PFE that you're using.

Default command key: None

Help User-Configurable Items

This is not a single **Help** menu item, but a list that you can define for yourself, so that you can access the help files that you want to work with quickly from PFE's own help menu.

PFE allows you to define up to five help files, and associate each with a menu item that is appended to the end of the **Help** menu.

To set up your own list of help files, you need to edit the **[help-files]** section of the initialisation file.

If you have not defined a specific context help file with the **context-help-file** key in the **[options]** section of the initialisation file, PFE will use the *first* file defined in the **[help-files]** section when you use the **Help Context Help** command.

Reference Not Available

Sorry, the material for this section is not yet complete

Procedures

This section of the help file contains overviews and descriptive material that will explain the features of PFE, and tell you the exact steps you need to take to achieve results.

In many cases, the descriptions will discuss running a dialog to perform an action. These topics will have links that take to step-by-step instructions for working with the dialogs; you can also access these instructions by clicking the **Help** button in the dialog itself, or by pressing the **F1** key.

Starting Up

Starting PFE

Running Single Or Multiple Instances
Specifying A Key Mapping File On The Command Line

Command Line Options

File Operations

Opening Existing Files From File Manager
Opening Existing Files From PFE
Creating A New File
Inserting An Existing File
Changing An Associated File Name
Saving A File To Disk
Closing A File
Mailing A File
Printing A File

Backing Up Existing Files
Working With Unix Format Files

Handling Text

Typing Text
Deleting Text
Selecting Text
Cutting, Copying and Pasting Text
Dragging And Dropping Text
Finding Text
Replacing Text
Text Processing Operations
Undoing Edit Actions
Templates
File And Window Modes

Moving About

Moving To Specific Lines
Finding Text

System Control

Window Manipulation
Screen Fonts

Starting PFE

Like any other Windows application, PFE can be started in a variety of ways.

- From **Program Manager** you can start it from an icon in a program group

- From **File Manager** you can start it by double clicking on its name. You can also start it up to edit one or more files by dropping their names onto the PFE program's name

You can also start it with the **File Run** commands of either of these applications; and from any other application that can launch programs.

However you start PFE, you can if you wish include on its command line a list of files that it is to open. If you do give a list, PFE will open each file in turn. Filenames given on the command line may include the normal DOS wildcard characters, and PFE will open all files whose names match. Files named on the command line will be opened for editing unless you also include the command line option "**/v**", when they will be opened in *read-only* mode.

By default, only one instance of PFE will run at a time. When you start a second one, it will pass control to the first, commanding it to open any files you've given on the command line.

If you wish, you can have PFE perform some file actions automatically as it starts up. The **auto-file** setting in the **[options]** section of the initialisation file can be used to either have PFE create an empty, unnamed window, or show you the dialog for opening files when you start it with no file names on the command line.

Running Single Or Multiple Instances
Specifying A Key Mapping File On The Command Line

Command Line Options

Running Single Or Multiple Instances

By default, PFE attempts to run only one instance of itself at any time. When you start it for a second time, the first instance is detected, and it is activated. Any file names that you gave on the command line are passed to this first instance across a DDE link.

In many cases, you'll find this a convenient way to work, as it reduces the clutter on your screen, and you don't have to worry about which version of PFE is editing what file.

However, if you're working on several things at the same time, it can be distracting to have unrelated files being edited in the same instance of the editor, so PFE gives you the choice of running multiple copies of itself when you want.

INITIALISATION FILE CONFIGURATION

The easiest way to set PFE up to run with multiple copies of itself is to edit the initialisation file. Including the line

```
run-mode=1
```

in the **[options]** section tells PFE always to start new instances of itself.

COMMAND LINE CONFIGURATION

You can also control how PFE starts by using command line options, which override the initialisation file values. If you specify the "**/m**" option, PFE will always start a new instance; if you specify "**/s**", it will always try to activate a previous instance.

You may find it convenient to set up several icons in your Program Manager groups with different command line options. For example, you could have a general icon with a command line of simply

```
pfe
```

to start PFE up for general editing. To edit your system files, you might have icons with command lines like:

```
pfe /m c:\windows\win.ini

pfe /m c:\autoexec.bat
```

and so on. These will start independent instances of PFE to edit these files, without getting in the way of any other editing work you're engaged in.

DDE CONSIDERATIONS

When you start PFE with the "**/m**" command line option, or if you don't use a command line option and have run-mode set to 1 in your initialisation file, the instance of PFE that starts will *not* run as a DDE server. This is so that any instance you start later with a "**/s**" switch will not activate this instance instead.

If you *want* an instance of PFE to run as a DDE server, you can either start PFE with a "**/s**" option (if run-mode is not set to 0) or use the **Options DDE Server** command to start any instance running a server.

Note that if you have more than one instance of PFE running as a DDE server, it's not predictable which of them will be connected to by another application. If you do use DDE, it's normally best to ensure that only one instance of PFE runs, or only one of several instances is acting as a server.

Specifying A Key Mapping File On The Command Line

By default, PFE will look for a file called **pfe.key** (if you're running the Windows 3 version) or **pfe32.key** (if you're running the Windows NT version) in your Windows directory when it starts, and will load the key mappings that it contains.

To load mappings from a different file on start-up, or to force PFE to use the built-in defaults described in this help file, you can use the "**/k**" command line option to name any key map file. For example, starting PFE with a command line

```
pfe /k c:\keymaps\uemap.key
```

would load the key mappings in the file **c:\keymaps\uemap.key**. If you have a key map file **pfe.key**, but want to use the built-in key mappings described in this help file instead, use the "/k" option and give the filename as the single character "**-**" (minus).

Note that when you use the "**/k**" option, the dialog started by the **Options Key Mapping** command will not save changed mappings to the file you specify, or to the standard key mapping file, by default. When saving you will need always to give an explicit file name.

Opening Existing Files From File Manager

If you have already started PFE running, you can pass files to it for editing from File Manager using the drag-and-drop technique. You can also do this from any other application that can act as a drag-and-drop server.

To open one or more files in this way, select them in the File Manager window; then click the left mouse button down while over one of the names and hold it down. Move the mouse cursor to anywhere in PFE's window and release the mouse button.

PFE will open each of the files in turn. They will all be opened such that you can alter them, except for files whose directory entries have the **read-only** attribute set, which will be opened in *read-only* mode.

Opening Existing Files From PFE

To start editing one or more existing files, use the **File Open** command. This starts a standard dialog that allows you to move around your directories to find the files you want to edit.

If you want to prevent yourself accidentally changing the files, click the **Read Only** box to tell PFE to open them all in *read-only* mode. Alternatively, you can use the **File View** command, which automatically does this. Whatever options you use, any file whose directory attributes include **Read Only** will always be opened in *read-only* mode.

Normally, the dialogs that open and save files will show you all the files that exist in each directory. You can arrange to set up your own selections by configuring the *filter list* that appears in the dialogs.

When each file is loaded, PFE looks at the *file type* to determine such things as the tab size to use, whether to treat the file as C language source, whether automatically to strip any final **Ctrl+Z** character, and so on. You can configure the settings it uses for different file types with the **Options Default File/Window Modes** command.

If you select a file that is already open, PFE will not load a new copy from disk. Instead, it will activate one of the windows that are currently showing the file.

PFE will examine the contents of the file specified to see if it is a template file. If it is, it will be attached rather than opened for editing.

PFE can open files that are stored in the normal MS-DOS format (using CR-LF bytes to mark the end of each line) or in Unix format (using a single LF byte) automatically.

Configuring The File Type Filters

Whenever you use one of the **File Open**, **File View**, **File Insert**, **File Save As** or **File Write** commands, PFE will show you a standard Windows dialog that allows you choose the filename to use, moving around all your directories.

The possible file names in the current directory are shown in the list on the left; the files shown in this list are determined by your selection from the **List Files of Type** list at bottom left.

By default, there will be only one item in the list, which selects all files, using a pattern of "***.***". If you wish, you can configure this list to let you select a smaller group of files: for example, if you program in C, you might wish to see only files with types of "**.c**" and "**.h**" in one group, and files of type "**.dlg**" in another.

The contents of the list are determined by the **[fileopen-filters]** section of the file **pfe.ini**, which is held in your Windows directory.

Creating A New File

The **File New** command opens a new, empty window in which you can type.

The window will not initially be associated with a file name. In order to save what it contains to disk, you'll need to use a command such as **File Save As**, which allows you to set the file name.

When the window is created, PFE applies the set of modes - that is, the tab size, wrap column, and other details - that are defined for "new files". You can use the **Options Current File/Window Modes** command to change them for the new window; or you can change the default settings using the **Options Default File/Window Modes** command.

Inserting An Existing File

To insert an existing file into the window that you're editing, place the caret to the point at which you want the insertion to be made, then use the **File Insert** command.

This starts a standard dialog that allows you to move around your directories to find the file you want to insert. PFE can insert files that are in normal MS-DOS format (using CR-LF bytes to mark the ends of lines) or in Unix format (using single LF bytes) automatically.

PFE will insert the file into the window, leaving the caret positioned after the last character inserted.

By default, the dialog will show you all the files existing in each directory, but you can configure it to show you a restricted set.

If the file to be inserted is a template file, PFE will reject the operation.

Changing A File Name

When you edit a file, PFE will associate the name of the file with all the windows that show it. When you then use the **File Save** command to write the file to disk, PFE will use the associated name when it opens the file.

Some windows - such as those that you've created with the **File New** command, or those holding DOS command output produced by the **Execute DOS Command To Window** command - will not have file names associated with them. Or you may want to change the associated name to something else.

You can change the file name associated with a window, and save it to disk at the same time, by using the **File Save As** command. This runs a dialog that allows you to specify a file name; after saving the file, that file name is associated with all the windows showing the same file.

Alternatively, you can change the file name without writing to disk by using the **File Name** command. This also runs a dialog that lets you specify the associated file name, but makes no changes to your disk.

Whatever method you use to change the name of a file, PFE will examine the name and change the file and window modes that apply to the file, and to the current window, to match the new file type.

Saving A File To Disk

To save a file to disk you have a choice of two commands.

- The **File Save** command will save the contents of the current window to disk, using the file name that's associated with the window. Any previous contents of the file will be lost.

- The **File Save As** command prompts you to supply the name of the disk file to write to. If you choose an existing file, PFE will warn you and ask if you really intend to overwrite it. The file name you give then becomes associated with the window.

If the window doesn't have a file name associated with it yet, both commands will ask for a file name. You can also associate a file name with a window at any time, without writing it to disk, using the **File Name** command.

If the current window contains a template, you won't be able to use the **File Save** or **File Save As** command. Instead, use the **Template Store** or **Template Store As** commands.

By default, you can't use the **File Save** command if you haven't changed the file. You can configure PFE to allow you to do this by placing the line

```
allow-save-always=1
```

in the **[options]** section of the initialisation file.

When saving a file, PFE will normally preserve a backup copy of any existing file of the same name, so that you can quickly revert to a previous version of your work.

Setting The Format of the Saved File
Backing Up Existing Files

Setting The Format of a Saved File

PFE can save files using two ways of marking the end of each line:

- **DOS Format** writes the file so that each line is terminated by a carriage-return byte and a line-feed byte. This is the normal format for DOS files, and you should use this if you want the file to be readable by other DOS applications.

- **UNIX Format** writes the file so that each line is terminated by only a line-feed byte. This is the format used in UNIX systems; however, the file probably won't be usable with other DOS or Windows applications.

The format that PFE will use to save a file is shown in the status bar.

There are several ways to set the format that a file will be saved in:

- By default, PFE will detect that a file is in Unix format when it is opened, and will set the save format to "Unix".

- You can change between DOS and Unix formats by double-clicking the left mouse button in the format area of the status bar.

- You can use the **Options Current File/Window Modes** command to change the format of the current file.

- You can use the **Options Default File/Window Modes** command to set the format for files of specific types.

You can also control two aspects of what happens to the final line in the file when it is saved:

- You can choose whether or not the last data character in the file is followed by a line terminator. By default, PFE will always write a line terminator.

- You can choose to have PFE automatically add a **Ctrl+Z** character after the file data has been written. (If the last character in the file is *already* a **Ctrl+Z** character, PFE will not add a second)

You can determine the settings used for the current file with the **Options Current File/Window Modes** command, and can set the default values used for files of given file types with the **Options Default File/Window Modes** command.

Working With Unix Format Files

Backing Up Existing Files

When you save a file to disk, it's often desirable to be able to keep a copy of any file of the same name that already exists. You can then quickly revert to a previous version of your work.

By default, all operations that write files to disk will perform a backup action. There are two ways that PFE will do this, which you can choose between by setting the **backup-mode** value in the **[options]** section of the initialisation file.

If you set the backup mode value to 0 (this is the default value), PFE maintains backup copies of files in the same directories as the files themselves. The exact steps taken are these:

1. If the file being saved exists already, PFE renames it. The new file name is the same as the original, with the file type changed to be "**.$$$**". Any file of this name that exists already will be overwritten

   If the file *already* has a file type of "**.$$$**", no backup is taken.

2. The data in the current window is then saved to disk

Then, if you need to revert to the previous version of the file, you will find it in the same directory as the original, with a file type of "**.$$$**"

If you set the backup mode value to 1, PFE maintains backup copies of files in subdirectories. The exact actions taken are these:

1. If it does not already exist, PFE will create a backup sub-directory in the same directory that contains the file to be backed up. The directory will be called "**$PFEBK**" by default; but if you wish it to have another name you can set this with the **backup-directory** statement in the **[options]** section of the initialisation file.

2. The file that would be overwritten is moved into this backup sub-directory, replacing any file of the same name that is already there.

3. The data in the current window is then saved to disk

Then, if you need to revert to the previous version of the file, you will find it in the backup sub-directory, with the same name as the original file.

If the file you're editing is actually in a backup sub-directory already, PFE will warn you of this before doing anything else. You will be given the choice of creating a further backup sub-directory as described above; or of saving the file without taking a backup; or of cancelling the operation.

You do need to take care that the process of taking a backup does not require the use of any pathname that exceeds the maximum supported by the operating system. Using the default name "**$PFEBK**" for the backup sub-directory, you need to ensure that the total size of the original file's pathname, plus 7 extra characters, does not exceed 65 for a FAT-based file system, or 255 for an NTFS file system. If your directories are deeply nested, you may need to define a shorter backup directory name with the **backup-directory** statement in the **[options]** section of the initialisation file.

Each backup mode has advantages and disadvantages.

- Backing up to files with a type of "**.$$$**" avoids creating lots of directories, but can clutter your source directory.

- If you edit files with similar names (such as **main.c** and **main.h**) the backup for *each* will be in **main.$$$**, so you will be able to revert only the file that was saved last. It's also not obvious which file is actually backed up.

- Backing up to subdirectories guarantees that each file is backed up to a unique file, and it's easy to associate a backup with an original. However, you can end up with a lot of subdirectories.

- Backing up to subdirectories keeps all your backup files in a single place, so you can delete them in one operation.

If you don't want to keep backup copies at all, you have several ways of telling PFE not to do so

- If you're using the **File Save As** or **File Write** commands, you can check or clear the **Keep Backup** checkbox to activate or deactivate backup

- You can use the **Options Current File/Window Modes** command to turn off backup for the file shown in the active window

- You can use the **Options Default File/Window Modes** command to turn off backup for files of given types, or for all files

- You can set the **backup-mode** value in the in the **[options]** section of the initialisation file to be **5**, which disables all backups regardless of the file modes you set

Of course, whichever mode you select, maintaining backup copies of all the files that you edit can consume a large amount of disk space. If you find that you are unable to save a large file, you can try to perform the operation with backup disabled.

Closing A File

When you've finished editing a file, you can close it with the **File Close** command. PFE will close all the windows that are showing the same file as the current window.

If you haven't yet saved some changes to the file, PFE will give you the opportunity to either write the file to disk before it is closed, or to abandon the operation.

If you want to close *all* the files that you are working on, use the **File Close All** command.

Closing the last, or only, window showing a file has the same effect as the **File Close** command.

If you're using PFE as an edit server for some other application, you may find it convenient to have the main window become an icon when you've closed all the files you've been working with. You can select this by including the line

```
minimize-on-empty=1
```

in the **[options]** section of the initialisation file.

Mailing A File

If you have a MAPI-compliant mail system, such as Microsoft Mail 3.0, installed on your system, you can mail files to other people directly from PFE.

To mail the file shown in the current window, simply select the **File Mail** command. PFE writes the data to a temporary file, and calls the mail system to send it as an attachment in a mail message.

If you aren't currently running the mail application, or if you aren't logged in to your mailbox, you'll see a dialog that asks you for your mail username and password. You'll then see the standard mail composition window, with the file you're sending shown in the message as the PFE icon, and you can add text or other attachments as you want.

If you're editing a file that already exists on your disk, this copy is *not altered* by this process.

Note that you'll need to have sufficient room on your disk for PFE to save the entire contents of the current window as a temporary file. This temporary file will be deleted once the mail process is complete.

Printing A File

You can print the file that is being shown in an edit window by using the **File Print** menu command. PFE will show you a dialog that lets you choose what part of the file you want to print: you can select to print either all of it, or only the part that you've selected, or the part between two line numbers.

You can specify the last line of the file as the word **end** in the rightmost of the line range edit controls.

The dialog box shows you the name of the printer that PFE is currently using - if you haven't specified otherwise, this will be your system default printer.

The file is printed using the same tab width that is being used in the window, and with line numbers if you have line numbering turned on. You can configure both these settings with **Options Current File/Window Modes** menu command.

You can quickly turn line numbering on or off by clicking the line number toggle button on the tool bar.

Setting Up The Printer
Selecting A Printer Font

Setting Up The Printer

To configure the printer that PFE is to use, use the **File Print Setup** menu command or click the **Setup** button in the dialog started by the **File Print** menu command. This will start the printer setup dialog.

The dialog lets you specify:

- The printer you wish to use
- What page margins are to be applied when you print on this device
- What font is to be used when you print to this device

You will also be able to run the printer's own setup dialog to set device-specific information.

All the settings that you make in this dialog are remembered by PFE on a per-printer basis, so you can have different settings for different printers.

The Printer Setup Dialog

Selecting A Printer Font

Clicking the **Fonts** button in the dialog started by the **File Print Setup** command starts a sub-dialog that lets you select the font to be used with the selected printer device.

Because PFE is a text editor rather than a word processor, you're restricted to choosing fixed pitch fonts, where the characters are all the same width.

PFE records the values you set in this dialog with the selected printer: the next time you choose this device, it will use the same settings.

The Printer Font Dialog

Working With Unix Format Files

Under MS-DOS, files are normally stored with the end of every line indicated by a **Carriage-Return** (CR) byte followed by a **Line-Feed** byte (LF). Most MS-DOS and Windows programs expect files to be in this format, and will not work correctly otherwise.

However, you may wish sometimes to handle files in Unix format, where lines are terminated by a single LF byte. You may have downloaded a file from a Unix system that needs conversion, or you may be using PFE to edit files on an NFS-mounted Unix file system, for example.

PFE is able to handle files in either MS-DOS or Unix format (or indeed, files using a combination of formats). By default PFE will detect that a file is in Unix format, and automatically set that format for when the file is saved.

The format that PFE will use to save a file in is shown in the status bar. You can quickly change the format by double-clicking the left mouse button in the appropriate area.

Opening Unix Format Files
Saving Files In Unix Format
Detecting Unix Format Automatically

Opening Unix Format Files

PFE handles opening and reading files in Unix format automatically, and you need take no special action to tell it that the file you are opening or inserting is a Unix file.

Depending on how have you have configured the optional file modes associated with the *file type* of the file being opened, PFE may set the **Save in UNIX format** mode on the file, so that when you write it to disk, lines will be terminated by single line-feed characters.

By default, files that are detected as being in Unix format when they are opened will be saved in Unix format.

Saving Files In Unix Format

PFE is able to write files to disk in either the normal MS-DOS format, using CR-LF as a line terminator, or in Unix format, using LF.

The format used to write a file is determined by one of the file modes - the **Save in Unix Format** mode - that you can associate with a file. You can set this mode in several ways:

- You can check the **Unix format** box in the dialog shown when you use the **File Save As** command. This will save the current file in UNIX format, and will also set the **Save in Unix Format** mode so that this becomes the default format when you write the file to disk again

- You can double click the left mouse button in the area of the status bar that shows the file format until it shows the word "**Unix**".

- You can check the **Save in Unix Format** box in the dialog started by the **Options Current File/Window Modes** command. This also makes Unix format the default for every time you write the file to disk

- You can have PFE set the mode automatically when the file is opened by using the **Options Default File/Window Modes** command. This lets you set the mode on either every file that you open, or only on those with specified file types

Detecting Unix Format Automatically

When opening a file, PFE looks at what characters are used to terminate a line. If the first line is terminated by a single Line-Feed character, PFE assumes that the file is in Unix format; otherwise, it will assume the file is in MS-DOS format.

Normally, if a file appear to be in Unix format when opened, PFE will automatically set the **Save in Unix Format** file mode, so that when you save the file it will be written in the same Unix format.

If you work with Unix files on your system, this will probably be what you want to happen. However, if you normally want to save files in DOS format, however they look originally, you can disable this feature by editing the initialisation file to include the line

```
auto-format=0
```

in the **[options]** section.

Typing Text

You can type text with PFE in the same way as for any other Windows application.

Normally, what you type appears on the screen, displacing other characters to the right if you're not at the end of a line. This is termed **Insert** mode.

If you prefer, you can switch to working in **Overwrite** mode, where a character you type will *replace* the one under the caret.

The mode is shown as either "**INS**" or "**OVR**" in the status bar. You can also tell by the shape of the caret: in insert mode it's a vertical line, and in overwrite mode it's a block.

You can switch between the two modes in several ways:

- Pressing the **Ins** key changes from one mode to the other
- Double clicking the left mouse button in the area of the status bar that shows the mode changes from one to the other
- You can change between the two modes, and also change other settings for the current window, with the **Options Current File/Window Modes** command
- You can set the default mode to use for files of specific types with the **Options Default File/Window Modes** command.

Remember that insert and overwrite modes apply to individual *windows*. You can have, for example, one window that shows a file in insert mode, and another showing the same file in insert mode if you wish.

Automatic Text Wrapping

Deleting Text

Pressing the **Delete** or **Del** keys will delete the character to the left of the caret, and move the caret left.

If you want delete an entire line or the last part of a line, there are two useful commands that let you do it in a single operation. The **Edit Delete Line** command deletes the *entire* line that the caret is in; the **Edit Delete To End Of Line** command deletes everything from the current position of the caret to the end of the line that contains it.

To delete a large area of text, you can select it with the keyboard or mouse and then press **Delete** or **Del**, or use the **Edit Cut** command.

Selecting Text

Like other Windows applications, PFE allows you to *select* an area of text and operate on it. The selected area appears highlighted.

You can select text using either a mouse, or with the keyboard. Also, the **Edit Goto Line** and **Edit Find** commands allow you to specify that all the text from where the caret starts to where it ends up is selected.

Selecting An Entire File

To select the whole of the file shown in the current window, use the **Edit Select All** command. PFE will highlight all the text in the file, and move the caret to after the last character.

Selecting A Single Word

PFE gives you a quick way to select a single word in your text.

- With a mouse, place the cursor anywhere within the word and double click the left mouse button
- Alternatively, or if you don't have a mouse, move the caret to anywhere within the word and use the **Edit Select Word** command

Selecting Using The Mouse

To select some text with the mouse, hold down the left button and drag the cursor. PFE will scroll the window in the appropriate direction when you move outside the window area. All the text that you select will be shown highlighted.

To select a single word, place the mouse pointer anywhere within it and double-click the left mouse button.

You can quickly select a large area by holding down the **Shift** key as you click the left mouse button. PFE will select all the text from the initial position of the caret to where the mouse cursor is. Since PFE does not move the caret within the file as you scroll with the up and down arrows in the vertical scroll bar, this can be a convenient way to select large areas.

If you have already selected an area, holding the **Shift** key down while you press the left mouse button *extends* the selection to the position of the mouse cursor.

Remember that, although you can have an area of selected text in each of the *files* you have open, only *one* window from any particular file can have a selection.

Selecting Using The Keyboard

As with other Windows applications, PFE allows you to select text using the keyboard.

Holding down the **Shift** key while using the **cursor keys** will move the caret up and down in the current window, selecting all the text that you pass over. Normally the caret moves one character or line at a time; holding the **Ctrl** key down at the same time as the left or right cursor key will move the caret by one word at a time.

Holding down the **Shift** key while pressing the **PgUp** or **PgDn** keys will scroll the window by one window's worth of lines, selecting all the lines that are passed over. Holding the **Ctrl** key down at the same time will move you to the start or end of the file respectively.

Holding down the **Shift** key while pressing the **Home** and **End** keys will move the caret to the start or end of the current line respectively, selecting all the text passed over.

To select a complete word, move the caret to anywhere within it and use the **Edit Select Word** command.

Cancelling A Selection

To cancel a selected area of text and return it to unselected state, you should either click the left mouse button once anywhere in the window concerned; or press the **5** key on the numeric keyboard with Num Lock off.

Cutting, Copying and Pasting Text

PFE uses the standard Windows *clipboard* for cutting, copying and pasting text. You can move text around in one edit window; move it from one window to another; or transfer text to and from other applications.

You can select the text with the mouse or keyboard, and then operate on it with menu commands. An alternative, very quick way of moving and copying text, is to use PFE's drag and drop facility.

You can work with very large amounts of data when cutting, copying and pasting. If you are using an 80286 based system, the maximum amount of data that can be handled is 80 bytes less than 1 megabyte. On an 80386 or higher, the maximum size is 16 megabytes less 64 kilobytes. With the Windows NT version of PFE there is effectively no limit.

However, you should normally work with small amounts of data, even if it means moving a large amount of text around in several operations instead of one. Clipboard data is stored by Windows in your system's main memory, and obviously the larger the amount you try to store, the more likely you are to run out of space. Also, because of the way Intel processors operate under Windows 3, PFE handles cutting, copying and pasting of more than 64 kilobytes of data significantly slower than smaller blocks.

Remember too that the clipboard is *not* emptied after you paste it into a file. This means that, when you exit PFE, you may still have a large amount of data stored in the clipboard taking up main memory space, which could cause you memory starvation problems when running other applications. You can reduce the amount of data stored in the clipboard by copying a single character to it before exiting from PFE; or you can use the Windows **Clipboard Viewer** application to empty the clipboard at any time.

Cutting Text To The Clipboard
Copying Text To The Clipboard
Pasting Text From The Clipboard
Dragging And Dropping Text

Cutting Text To The Clipboard

To cut some text from the current window and place it on the clipboard, you should first select it, then use the **Edit Cut** command.

PFE will delete the selected text from the edit window after making a copy of it in the Windows clipboard. Anything that was previously held in the clipboard will be lost.

Copying Text To The Clipboard

To copy some text from the current window and place it on the clipboard, you should first select it, then use the **Edit Copy** command.

PFE will duplicate the selected text in the clipboard, and leave the window unchanged.

By default, PFE leaves the selected text highlighted after this operation. If you prefer to have the highlighting removed, you should place the line

```
deselect-on-copy=1
```

in the **[options]** section of PFE's initialisation file.

Pasting Text From The Clipboard

When the Windows clipboard contains some ASCII text (either put there from PFE or some other application) you can paste it into the file you're editing at the position of the caret with the **Edit Paste** command.

If you have selected any text prior to using this command, PFE will delete it before performing the paste operation.

Dragging And Dropping Text

If you have a mouse, a very quick way to move text around within a window is to use PFE's *drag-and-drop* facility, which lets you simply pick up text from one place and drop it where you want it.

To use drag-and-drop, you should do this:

1. Select the text you want to move or copy to another part of the window

2. Place the mouse cursor within the selected area. The cursor shape changes from an I-beam to an arrow to show that you can use drag and drop.

   Press the left mouse button and hold it down. The cursor changes to the *drag-and-drop* cursor. If you want to *copy* the selected text rather than move it, you should also press and hold down the **Ctrl** key

3. Move the cursor to where you want to move or copy the text; the caret shows you the exact place where the text will go. Moving the mouse cursor outside the window will cause it scroll in the appropriate direction

4. When the caret is at the required location, release the left mouse button (if you're copying the text, make sure you hold the **Ctrl** key down until *after* you release the button).

If you decide that you didn't really want to carry out the operation once you've started it, release the **Ctrl** key if you're holding it down, and press the **ESCAPE** key. The mouse cursor will return to its normal shape, and you can now release the mouse button without changing the file.

Note that the selected text is *not* copied to the clipboard when you perform a drag-and-drop; anything you have in the clipboard already will be unaffected.

Normally, PFE will *move* the selected text unless you hold the **Ctrl** key down when releasing the mouse button. If you prefer, you can change this so that text is *copied* by default, and *moved* only if the **Ctrl** key is down. To do this, edit the initialisation file to include the following line in the **[options]** section

```
dragdrop-flip=1
```

Depending on how responsive your system is, and how you like to work with an editor, you may prefer to have the drag and drop facility turned off. To do this, edit the **pfe.ini** file in your Windows directory to include the line

```
use-dragdrop=0
```

in the **[options]** section. If drag and drop is not available, the mouse cursor does not change shape when it is over highlighted text.

Finding Text

To search for an occurrence of some text in a window, place the caret at the point where you want the search to start, and use the **Edit Find** command. This starts a standard Windows dialog that lets you specify what you want to look for, and other details.

The dialog box remains visible after you've made the search, enabling you to repeat it in the same or in another edit window.

The Edit Find Dialog

Finding Special Characters
Repeating A Find Operation

Finding Special Characters

When you're searching you may want to look for occurrences of characters that you can't type in the edit control of the Find dialog. PFE lets you specify them using a special notation:

\f	represents a Form Feed character
\n	represents the end of a line
\t	represents a TAB character
\\	represents a single '****' character

To specify an arbitrary character code you can use the notation "**\xnn**", where "**nn**" is two hexadecimal digits. The null value "**\x00**" is not permitted.

Repeating A Find Operation

There are two ways of repeating a text search. Because the Find dialog remains visible until you explicitly dismiss it, you can repeat a search - either exactly or with changes - simply by pressing the **OK** button. You can perform the search in any window.

However, on standard VGA screens especially, the dialog can take a large amount of space, so you may prefer to dismiss it and use the **Edit Repeat Last Find** command to repeat a search. This command repeats the last search operation you did *exactly.*

Replacing Text

To search for an occurrence of some text in a window, and replace it with some different text, place the caret at the point where you want the search to start, and use the **Edit Replace** command. This starts a standard Windows dialog that lets you specify what you want to look for, what to replace it with, and other details.

The dialog box remains visible after you've performed the operation, so you can repeat the operation in the same or in another edit window.

The Edit Replace Dialog

Replacing Special Characters
Repeating A Replace Operation

Replacing Special Characters

When you're replacing text you may want to look for or insert characters that you can't type in the edit control of the Replace dialog. PFE lets you specify them using a special notation:

\f	represents a Form Feed character
\n	represents the end of a line
\t	represents a TAB character
\\	represents a single '****' character

To specify an arbitrary character code you can use the notation "**xnn**", where "**nn**" is two hexadecimal digits. The null value "**x00**" is not permitted.

Repeating A Replace Operation

There are two ways of repeating a text replacement operation. Because the Replace dialog remains visible until you explicitly dismiss it, you can repeat an operation - either exactly or with changes - simply by pressing one of the **Find Next**, **Replace** or **Replace All** buttons. You can perform the operation in any window.

However, on standard VGA screens especially, the dialog can take a large amount of space, so you may prefer to dismiss it and use the **Edit Repeat Last Replace** command to repeat the process. This command repeats the last replace operation you did *exactly.*

Text Processing

Although PFE is not intended to be used as a word processor, it does support some text processing features that are of use to those editing program source, and a few of general applicability.

Automatic Text Wrapping

When you're typing ordinary text, rather than a program source, it's often convenient to have the program you're using handle the business of fitting the text into the available line width.

The quickest way to activate text wrapping for a window is to double-click the left mouse button in the wrap area of the status bar. This will show the text "**No Wrap**" when wrapping is not active. The double click action will turn wrapping on, using the currently set wrap column, which by default is column 72.

You can also switch wrapping on and off, and also define the wrap column, for the current window, using the **Options Current File/Window Modes** command.

You can also cause text wrapping to be activated automatically for all files of a specific type by using the **Options Default File/Window Modes** command.

Whether wrapping is active, and the wrap column used, is set on a per-window basis. You can, if you want, edit the same file in two different windows with different wrap settings.

In any window for which wrapping is active, PFE monitors as you type, and whenever you type a non-space character at a screen position to the right of the defined wrap column, it will automatically break the line you're typing in at a suitable place, moving some text and the insertion point down to the next line.

The points at which PFE will break a line vary with the *language type* defined for the window with the **Options Current File/Window Modes** command.

Language type "(none)" or "C"

The line will be broken at the closest white space character or hyphen to the defined wrap column.

Language type "TeX"

The line will be broken at the closest white space character to the defined wrap column

Note that PFE will wrap the line you're typing *only* when you're typing at the right hand end of it. If you move the caret to somewhere within the line and type, the line won't be wrapped even if it extends past the wrap column.

PFE does not perform any sort of text justification.

Indenting And Undenting Text

When you're editing program source, you'll very often want to change the indentation of some or all of it. PFE gives you a powerful facility to change the indentation level of either a single line, or of a larger block of text.

To indent a *single* line, place the caret anywhere within it and use the **Edit Text Indent** command. PFE will move the entire text of the line to the right by one tab stop, inserting either a single tab character or a number of spaces, as appropriate for the window. Lines that are empty are not affected.

To undent a *single* line, place the caret within it and use the **Edit Text Undent** command. PFE will move the text left by one tab stop: if the line starts with a tab character, this will be deleted; otherwise, PFE removes as many space characters as it can up to the width of a tab stop. A line that does not start with either a tab character or a space will not be altered.

If you want to indent or undent more than one line, the procedure is similar; but now you indicate the lines you want to change by selecting them. Now when you use either of the commands, PFE will alter all the lines that are selected. Unlike most operations that affect selected text, indenting and undenting operates on the whole of any line containing a highlight, and not just on the highlighted text itself.

Changing Case

PFE lets you quickly change the case of the text that you've typed. First, you should select the text whose case you want to change.

The **Edit Text Uppercase Selection** command will make all the characters you've selected upper case; the **Edit Text Lowercase Selection** command makes them all lower case.

Inserting Characters by ASCII Code

PFE allows you to insert characters with arbitrary ASCII codes between 1 and 255 that you cannot type on the keyboard. This facility permits you to insert control characters such as **ESC** and **Form Feed**, or characters with codes greater than 127.

The **Edit Text Insert ASCII Code** command starts a dialog that allows you to specify the ASCII code to insert.

Handling Braces

PFE provides several commands that allow you to manipulate *brace characters* such as brackets and parentheses, which often delimit sections of text or of program sources.

The definition of a brace character varies, depending on what language type you have set in a window's window modes. If you have defined the language type to be "**C**", braces are defined as the characters that have that syntactic meaning in the language - opening braces are **{ [** or **(**, and closing braces are **}]** or **)**.

If you have set the language type to be "**none**", brace characters are defined as those commonly used in text or many other languages - opening braces are **{ [(** or **<**, and closing braces are **}])** or >.

Two commands allow you easily to locate braces that match each other. If the caret is positioned on either an opening or a closing brace character, the **Edit Text Match Brace** and **Edit Text Match Brace Select** commands will move the caret to the matching one, having regard for nesting of braces. The latter command will also highlight all the text between and including the brace characters.

The **Edit Text Widen Brace Select Command** command helps you locate and manipulate brace-delimited sections of text. If the caret is positioned anywhere between two matching braces, the command will move the caret to the opening brace character and select the text between and including them. The next use of the command will widen the selection to include the brace characters that completely enclose this selected area, and so on. The result is that, in C language source, for example, you can select larger and larger syntactic elements with simple repeated keystrokes, and copy, delete or otherwise manipulate them as you choose.

In locating matching brace commands, PFE's actions are again moderated by the defined language type. For a language type of "**none**", all characters are examined, and there is no syntactic meaning to any text.

For a language type of "**C**", however, the search is made with regard to C language syntax. Brace characters within comments, delimited strings or pre-processor directives are disregarded, as are those defined as character literals. In most cases, PFE will be able accurately to match a brace character. However, some complex syntax may mislead it, especially if the syntax is invalid or incomplete.

Undoing Edit Actions

PFE includes a powerful and configurable system that will let you correct your editing mistakes easily. The edit changes you make are recorded in memory, and you can reverse them with the **Edit Undo** command, which can be initiated quickly by clicking on a **tool bar** button.

When recording the edit changes you make, PFE handles them as a sequence of *actions*. By default it will record the last 32 actions performed on each open file; but you can change this to any number between 8 and 128 by setting the **max-undo-actions** value in the **[options]** section of the initialisation file.

In most cases, it is obvious what an undoable *action* is. For example, deleting a line with the **Edit Delete Line** command is a single action, as is pasting from the clipboard with **Edit Paste** or inserting a template with **Template Insert**.

However, typing and deleting single characters is handled somewhat differently. PFE does not regard each keypress as an *action*: rather, it stores as one action all the keystrokes you make, until you begin doing something else. Undoing the action then reverses the whole collection of keystrokes, rather than just the last one.

For example, simply typing characters accumulates keystrokes in a *typing* action; deleting with the **BackSpace** key accumulates keystrokes in a *delete backwards* action, and deleting forwards with the **Delete** key accumulates them in a *delete forwards* action. PFE only stops accumulating keystrokes in an action when you do something *different*:. For example, PFE would stop accumulating keystrokes in a *typing* action if you began to delete characters; and it would stop accumulating keystrokes in a *delete backwards* action if you began typing characters again.

Accumulation of keystrokes will also stop if you use any key that moves the caret; move the caret with the mouse; or perform any command.

Obviously, recording the edit changes you have made to a file consumes memory; and if you have specified that PFE should record a large number of edit actions, and cut and paste large amounts of data, you may use up very large amounts indeed.

PFE provides several techniques to keep memory usage under control:

- The **Edit Clear Undo Actions** command lets you discard all the recorded actions for the current file, freeing off the memory used, at any time. Of course, once you have used this command, you can no longer reverse any changes.

- When you write a file to disk, PFE will by default clear the recorded actions for the current file for you, releasing the memory used. If you wish to suppress this, and retain all recorded actions until you decide to clear them yourself, you can set the **save-clears-undo** value in the **[options]** section of the initialisation file.

Templates

Templates are pre-built sections of text that you can include into your files, saving you from repeatedly typing the same things.

For example, you might like to document your C procedures by preceding each one with an explanatory section like this:

```
/************************************************

   start_editor
   ------------

   This procedure starts the system in edit
   mode

************************************************/
```

Rather than type the lines of asterisks every time, you could set up a *template* containing the text:

```
/************************************************


<???>
    -------------

************************************************/
```

and insert it automatically in a simple operation. The "**<???>**" characters are a *template mark* to show where you have to type in extra items (such as the name of the procedure in this example). PFE lets you move between these marks with a single keypress, so that editing is very simple and convenient.

Templates are held in *template files*, which you can regard as a sort of library. You can have as many template files ready for use, or *attached* as you want, and you can insert templates from any of them at will. Template files are not ordinary text files and can't be edited directly, but the **Template** menu provides you with options for creating template files, attaching and detaching them, creating, editing and deleting templates.

Each template file can hold a maximum of 64 templates. The total space taken up for data in a template file cannot exceed a total of approximately 62 kilobytes. Each line in a template requires one byte for each character, plus two extra bytes; and one further byte is required for each template.

You can attach template files manually at any time as you need them. However, you'll probably find that you create a set of templates that you use all the time: you can arrange for PFE to attach these automatically when it starts up. You can also have a set of templates that are specific to a particular project, and this also can be attached automatically.

Creating A Template File

Before you can store a template you must have a *template file* to place it in. The **Template Create File** command starts a standard dialog that lets you specify the name of the file you want to create. PFE will create the file you specified and write a small amount of binary control information into it.

You should use the file type **.tpl** when you create a template file. You should not try to template files directly, as they do not contain normal text.

When you create a template file it is automatically attached and ready to use.

Attaching A Template File

Before you can access any of the templates contained in a template file, you must first *attach* it. PFE loads the file into memory and constructs an index to the templates it contains, after which you can manipulate it.

You attach a template file with the **Template Attach File** command. This starts a dialog that lets you select the file you want to attach.

If you wish, you can attach a template file in *read only* mode. You will be able to insert templates contained in this file into files that you are editing, but you will not be able to add new templates to it or edit existing ones within it.

Having a template file attached uses some memory: PFE uses global memory to hold the data within the template file, and some control blocks. You can recover the memory used by an unwanted template file by *detaching* it with the **Template Detach File** command.

Automatically Attaching Template Files

Automatically Attaching Template Files

If you like, you can have PFE attach some template files automatically when it starts up, rather than attaching them manually one by one with the **Template Attach File** command. You can specify the files in two ways:

- As it starts, PFE scans the current directory for a file called **auto.tpl** and attaches it if it is a valid template file

- You can specify a template directory in PFE's initialisation file with a line in the **[templates]** section of the form

  directory=pathname

  When it starts, PFE will look for a directory with the specified pathname, and will attach *all* the files it contains with a file type of **.tpl**

Creating A Template

To create a template, use the **Template New** command. This opens up an edit window that is exactly the same as any other; the sole difference is that the window caption will show that you're editing a template.

You can edit inside the window exactly as in any other. You have an additional facility, though: you can use the **Template Insert Mark** command to insert a *template mark*. This is a character sequence (actually the characters **<???>**) that are automatically searched for by the **Template Find Mark** command. You can use template marks to hold the position for things to be filled in once you've inserted a template into your text file.

In order to use the template once you've finished editing it, you must store it in a template file. You can create a template at any time, but to store it you must have a template file attached. **Template Store As** command starts a dialog that shows you all the template files that you've attached. You should select the file you want to store the template in, then type the template's name. This can be any sequence of up to 16 characters, used to identify this particular template.

PFE will write the template information into its memory copy of the template file, and you can then insert the template into files at will. However, the template will *not* be saved to disk until you use the **Template Save File** command.

Inserting A Template

At any time when you're editing, you can insert a template from any attached template file at the position of the caret.

The **Template Insert** command shows you a dialog that gives you a list of all the attached template files. When you select one of the files you'll see a list of all the templates that it contains; you can then select the particular template that you want to insert.

PFE will read the template data from the template file and insert it into the current window at the position of the caret. Unlike the case when you're inserting another file or pasting from the clipboard, the caret remains at the *start* of the inserted data. This allows you to use the **Template Find Mark** command to move automatically to any template marks that you've put in the template to show where extra text needs to be inserted.

Editing A Template

In order to change a template that you've stored in a template file you need first to attach the file, then use the **Template Edit** command. Since template files do not contain plain text, you shouldn't attempt the edit them directly.

The command shows you a dialog that lists all the attached template files. Selecting one of the files shows you a list of all the templates it contains, and then you can select the template you want to edit.

When you've selected a template, PFE will place it in an edit window for you. The window is exactly the same as one for editing a normal text file, but the caption will show you that it's actually a template you're editing. You can edit the template in any way you like.

Once you've finished your edit, you can update the template file with the **Template Store** command. This will delete the old version of the template from the template file and replace it with the new one. If you want to save the contents of the edit window as a template with a different name, or into a different template file, use the **Template Store As** command.

A template *file* is *not* updated on disk when you finish editing a template. To preserve your changes you need to explicitly save it using the **Template Save File** command.

Because a template is not regarded as an ordinary file, you can't use the **File Save** or **File Save As** commands. If you need to save the template into an ordinary file, use the **Edit Copy** command to copy the data to the clipboard and paste it into a new window.

When you're editing a template, you can use the **Template Insert Mark** command to insert a *template mark*. This is a character sequence (actually the characters **<???>**) that are automatically searched for by the **Template Find Mark** command. You can use template marks to hold the position for things to be filled in once you've inserted a template into your text file.

Deleting A Template

To delete an unwanted template from a template file, you must first attach it, then use the **Template Delete** command.

This starts a dialog that allows you to specify one or more templates in a template file and delete them. The template *file* is *not* updated on disk when you delete a template: to preserve your changes you need to explicitly save the file using the **Template Save File** command.

Saving A Changed Template File

When you edit a template, create a new one, or delete one, PFE makes the changes only to the copy of the template data that it keeps in memory. The template file on disk is not altered in any way.

To make your changes permanent, you need to use the **Template Save File** command This starts a dialog that shows you the names of all the template files that have been altered, allowing you to select those you want to write back to your disk.

Detaching A Template File

If you don't want to use the templates contained in a template file any more, you can regain some memory by removing the in-memory copy of the file with the **Template Detach File** command.

The command will start a dialog that lets you choose the template files to detach. Once you've detached a file, you won't be able to use the templates it contains until you attach it again.

PFE will not let you detach a template file if you're currently editing any of the templates it contains.

File And Window Modes

PFE allows you to change several aspects of how it operates in any particular window, by setting various *modes*.

> **File Modes** apply to files. If you're editing a file in more than one window, each will use the same file modes; for example, if a file has the mode **Save in Unix format** set, then PFE will save it in Unix format, whichever window is active when you do the save.
>
> **Window Modes**, in contrast, apply to individual windows, and you can have different modes set for each of the windows showing a file if you choose. Window Modes tell PFE things about how to present the text to you - they dictate how wide a tab stop should be, whether text is automatically wrapped at a right margin, whether line numbers are shown, and so on.

You can alter the Window Modes applying to a single window, or the File Modes applying to a file shown in a window, at any time. A dialog lets you set all the modes in one go; and in addition, there are shortcut ways of setting some very common ones.

You can also define sets of File Modes and Window Modes that PFE should automatically apply when it opens files. For example, you can arrange that PFE will edit all files with type "**.c**" and "**.h**" with a tab size of 4, while all files of type "**.txt**" are to be edited with a tab size of 8 and automatic text wrapping at column 72.

Setting Modes For A Window Or File
Setting Default File and Window Modes

Setting Modes For A Window Or File

If you want to change the Window Modes for the current window, or the File Modes for the file that's being shown in it, use the **Options Current Window/File Modes** command, which starts a dialog that lets you alter all the settings.

The dialog box reminds you of the name of the current file at the top. You should set the modes that you want by checking or unchecking the boxes, then press **OK** to action them.

Setting Default File And Window Modes

If you want, you can associate sets of File Modes and Window Modes with specific file types, so that when you open a file with a given type, or change a file's name to have a different type, it will automatically set the modes you've defined.

To define sets of modes, use the **Options Default File/Window Modes** command. This starts a dialog that lets you set up the information.

You can set the file and window modes that PFE is to apply

- When it creates a new file with the **File New** command
- When you use the **File Open** or **File View** commands to open an existing file of a given type, or change the name of the file associated with a window to one with a given type
- When you use the **File Open** or **File View** commands to open an existing file of any other type, or change the name of the file associated with a window to one with any other type

Moving To Specific Lines

You have several ways of moving to a specific line in a file. You can use the normal movement keys and watch the file position in the status bar at the bottom of the main window, or you can turn on file line numbers and use those to guide you.

Alternatively, you can use the **Edit Goto Line** menu command. This starts a dialog that lets you enter the line number directly, and will adjust the window so that the caret is on the first character of the specified line.

Window Manipulation

PFE uses the standard Windows Multiple Document Interface to allow you to edit as many files as you wish simultaneously. You can use all the normal MDI techniques to manipulate the windows, and there are some further facilities specific to PFE.

Selecting A Window

You have several ways of moving from one edit window to another.

The **Window Next** command moves you between windows that are not currently icons - any iconic windows are not restored by this.

With a mouse, you can click on any window to activate it. To prevent you accidentally moving the caret when you activate a window with the mouse cursor within the edit area, PFE will not change the caret position until you click a second time.

The **Window** menu contains one item for each of the first 9 edit windows; clicking the item that identifies the window will activate it (if the window is an icon, it will be restored). If you have more than 9 windows open, you'll see a **More Windows** item, which will run a dialog that will list all the existing windows.

Closing A Window

To close a window, you can use select the **Close** item from its system menu, double-click in the system menu box, or use the **Window Close** command.

If the window is the only one currently showing a file, PFE will act as if you'd used the **File Close** command and close the file also; if you have made any changes to the file you'll be prompted to see if you want to save them, discard them, or cancel the operation.

Moving And Sizing A Window

You can iconize any edit window by clicking on its minimize box, or via its system menu. You can also iconize *all* the edit windows that exist in one operation with the **Window Iconize All** command.

You can maximize a window so that it fills the full extent of the main window by clicking its maximize box, or via its system menu. You can also double-click anywhere on the window's caption bar to do this.

You can move a window around or resize it with the mouse, or via the window's system menu.

Duplicating A Window

PFE allows you to have as many windows as you want showing the *same* file - for example, you might find it convenient to have one window showing the start of a file, and another showing the end of it. You can also apply different window modes to the windows for various effects.

The **Window Duplicate** command will make an *exact* duplicate of the current window. PFE will create a second window that is the same size as the original, with all the same window modes, showing the same text.

You can edit the file concerned in any of the windows that are showing it, but there will be a small performance penalty, as PFE needs to synchronise a change made in one window with all the others.

Although you can select text freely when you have multiple windows on a file, only one of the windows can have an active selection at any time. If you select some text and then activate another window on the same file, the selection will be cleared.

Widening A Window

Often you may want to make a window as wide as possible, so that you can see the maximum possible amount of text in each file line. You can achieve this in the normal way that Windows allows you to resize a window; but an easier way is to use the **Window Widen** command.

PFE will automatically make the current window as wide as possible, moving it to the far left of the main window. The window's depth and vertical position are not affected.

Tiling Windows

PFE allows you automatically to position the windows you're working with to make best use of the available screen area. You can tile windows either *vertically* or *horizontally*.

Horizontal Tiling

The **Window Tile Horizontal** command arranges your windows in columns, so that you can see as many lines in each as possible.

Vertical Tiling

The **Window Tile Vertical** command arranges your windows one above the other on the screen, so that you can see as much of each line in as many windows as possible. Depending on the size of your display, only 2 or 3 windows can be arranged in this way without becoming too small to use; excess windows are tiled in a horizontal pattern in the lower part of the screen.

Windows holding the output from commands run by the **Execute DOS Command To Window** command are preferentially arranged at the top of the screen.

In each of the tiling cases, PFE will arrange any window icons neatly towards the bottom of the screen, and will always keep the bottom row of icons visible. When tiling vertically this can reduce the usable size of the screen by too much, especially if you have more than three non-iconic windows.

You can tell PFE to use the *full* depth of the screen for vertical tiling by including the line

```
max-vertical-tile=1
```

in the **[options]** section of the initialisation file.

Screen Fonts

By default, PFE will use the fixed pitch **System** font to display text in edit windows. However, it will allow you to use *any* fixed pitch font that exists on your system, as well as the standard ANSI and OEM fonts.

Clicking on the **Options Screen Font** menu command will show you a popup menu containing some screen-font related items.

To use one of the standard system fonts, click on one of the **System**, **ANSI** or **OEM** items. These will select the appropriate standard font.

To use a TrueType font, or one provided by some other system such as Adobe Type Manager, click on the **Other** item to start a standard dialog that will let you select from all the fixed-pitch fonts installed on your system

Selecting a screen font has no effect on any fonts you may have selected for printers.

The Status Bar

PFE normally shows you a **status bar** at the bottom of its window. This gives you several items of useful information, and also gives you a quick way of changing some settings and executing a few commands.

The status bar shows you:

- The line number and column number where the caret is. Double clicking the left mouse button in this area executes an **Edit Goto Line** command.

- The total number of lines in the file.

- Whether the file has changed : you will see a **#** if it has.

- Whether the file is *read only* : the characters **RO** will show if it is, and **WR** will show if you can alter it. Double clicking the left mouse button in this area will change between the two settings.

- The command sequence prefix key - that is, whether you've typed an active prefix key and PFE is waiting for you to supply the second key of the sequence.

- Whether your keystrokes and menu selections are being recorded in a keyboard macro. Double clicking the left mouse button in this area will turn recording on and off.

- Whether PFE will automatically wrap lines that you type, and the column number past which this wrapping will take place. Double clicking the left mouse button in this area will turn wrapping on and off.

- Whether the file will be saved in DOS format or Unix format. Double clicking the left mouse button in this area will change between the two formats. You cannot change the save format of a file that is marked as *read only*.

- Whether typing is to be *inserted* into what's there already, or will *overwrite* it : you see **INS** or **OVR** respectively. Double clicking the left mouse button in this area will change between the two settings, as will pressing the **Ins** key.

- The state of the **Num Lock** and **Shift Lock** keys

You can turn off the status bar and make a little more room for showing the files you're editing, or turn it on again, with the **Options Status Bar** command. On slower machines you will probably see a noticeable performance increase when you have the status bar turned off.

The Tool Bar

Besides controlling PFE from the keyboard, or by using the menu, you can also use the *tool bar* to make access to the most common operations simple.

The tool bar is a collection of coloured push buttons. By default it appears at the top of the screen below the menu, but you can place it at any of the main window's edges, or have it floating as a separate window.

You can invoke the function associated with a tool bar button by clicking the left mouse button when the mouse pointer is within it. Some buttons have more than one function, and you invoke these by holding either the **Shift** key, the **Ctrl** key, or both, down as you *release* the mouse button.

When you click the left mouse button on a toolbar button, PFE will show some text explaining its use in the status bar at the bottom of the screen. If you decide you don't want to perform the function, move the mouse pointer out of the tool bar area before releasing the mouse button.

The topic **Description of the Tool Bar Buttons** below describes each of the buttons fully. Note that in this help file, the buttons are drawn as they appear by default on screens with 1024 x 768 resolution. Some buttons may appear slightly differently on other screens.

You can also obtain help on the function of any button in the tool bar by using the **Edit Screen/Menu Help** command, positioning the question-and-arrow cursor on the button, and clicking the left mouse button once.

Not all of the tool bar buttons will be usable at all times: for example, the **Print** button can't work if you don't actually have a file open. The buttons that you *can* use will display icons in colour; those that aren't currently available will show grey shadow images.

Description of the Tool Bar Buttons
Moving the Tool Bar
Setting The Tool Bar Size

Setting The Tool Bar Size

Normally, PFE selects the bitmaps it draws for the tool bar buttons to match the resolution of your screen, so that the tool bar appears approximately the same physical size on all screens. However, if you have a large screen with a high resolution, you may want to force PFE to use a smaller toolbar for a better appearance.

You can override the automatic sizing of the tool bar by setting the **toolbar-size** value in the **[options]** section of the initialisation file. The option allows you to force PFE to use a small, medium or large set of tool bar buttons.

Moving The Tool Bar

By default, the tool bar appears as a row of button at the top of the screen below the menu. You can choose to place it at any of the main window's edges, or you can have it appear as a small floating window that you can place anywhere at all in the edit window. You can also, of course, choose to turn it off completely.

One way of moving the tool bar is to use the menu. The **Options Tool Bar** command gives you a list of possible options. You can *hide* the tool bar or *show* it, or specify where you want it to be placed.

If you have a mouse, you can also *drag* the tool bar to where you want it. You need to do this to start:

- If the tool bar is at one of the window edges, place the mouse pointer within it but *not* over one of the buttons. The cursor will change to the shape of a hand. Press the left mouse button and hold it down. The cursor shape now changes to a four-pointed arrow.

- If the tool bar is a floating window, put the mouse pointer over the caption bar at the top, press the left button and hold it down.

Now *drag* the tool bar to where you want it to be placed and release the left mouse button. If the mouse pointer is close to one of the main window's edges, the tool bar will be placed on that edge; if the mouse pointer is nearer the middle of the main window, the tool bar will turn into a floating window.

PFE will remember where you've placed the tool bar, and will put it there the next time you start the program.

Key Mapping

PFE allows you to change the details of what keys you press to perform most of its operations. For instance, you could set the **Ctrl+A** key so that pressing it executed the **Template Attach File** command if you found that convenient.

The operations that PFE performs are called *functions*, and these have fairly descriptive names such as **FileOpen** and **EditTransposeCharacters**. Most of the functions are similar in name to the commands used to control PFE over a DDE link, and to those on the menus.

The relationship between a key on the keyboard and the function that is performed by pressing it is known as a *key mapping*. You can have as many keys as you want mapped to a function.

You can store your personalised key mappings so that they are automatically applied every time you start PFE. You can also save them in key mapping files, so that different people using your system could load their own mappings; or so that you could have different mappings for different purposes

What Keys Can Be Mapped

PFE is controlled from the keyboard using two different types of key sequences. You can map almost all the available key combinations, so you should be able to find ones that suit your way of working.

Firstly, PFE uses *single-character* key presses such as **Ctrl+A**, **Shift+BackSpace**, **Alt+X** and **F7**, in a similar way to a standard Windows application. If you wished, you could set up PFE to operate entirely with such characters; this is how it will work by default.

The second mode uses *two-characters sequences*, where you first press a *prefix key*, and then press a second key. This form is similar in approach to the **EMACS** editor or the **WordStar** word processor. By default prefix keys are not used, but you can activate any or all of **Esc** key and **Ctrl+A** to **Ctrl+Z** as prefixes. A prefix key can be followed by any valid single-character key, so the number of mappable key combinations you have available is vast.

PFE does not allow you to remap some of the keys that have standard functions within Windows. Thus, you can't map **Alt+Esc** or **Alt+Tab**, for example, as these are the standard way of moving between task windows.

If you map keys such as **Alt+F** or **Alt+W**, they will no longer function as menu hot keys to drop down the appropriate menus. However, you can access the menus by pressing and releasing the **Alt** key alone, and using the cursor keys to navigate to the required menu.

Changing Key Mappings

To change the mappings of keys to functions to be as you want, use the **Options Key Mapping** menu command. This starts a dialog from which you can choose your requirements.

Defining and Changing Key Mappings
Changing The Active Prefix Keys
Saving A Key Mapping
Loading A Key Mapping

Defining and Changing Key Mappings

To map a key that is currently unmapped to a function, select the **Options Key Mapping** command, which starts the key mapping dialog.

Saving A Key Mapping

While you are running the key mapping dialog, you can choose to save your new mapping at any time.

If you used the "**/k**" command line option to load a specific key map file on start-up, you will need always to use **Save As** button and name the file you want to save the changes to explicitly.

If you didn't use the "**/k**" option, you'll still be able to use the **Save As** button to write your changes to a specific file. However, you'll also be able to use the **Save** button to save the changes to the **pfe.key** file in your Windows directory, which PFE will by default load when it starts up.

Loading A Key Mapping

PFE allows you to save as many different sets of key mappings as you wish, so that different people using your system can each have their own favourite settings; or you can have different sets of your own for different purposes.

Key mappings are stored in files with type "**.key**". When it starts up, PFE will automatically try to load the key mappings from **pfe.key** (in the Windows 3 version) or **pfe32.key** (in the Windows NT version) in your Windows directory unless you suppress this on the command line as described below.

FROM THE COMMAND LINE

To load a set of key mappings other than the default when PFE starts, you can use the "**/k**" command line option to name any key map file. For example, starting PFE with a command line

```
pfe /k c:\keymaps\uemap.key
```

would load the key mappings in the file **c:\keymaps\uemap.key**. If you have a standard key map file in your Windows directory, but want to use the built-in key mappings described in this help file instead, use the "/k" option and give the filename as the single character "**-**" (minus).

Note that when you use the "**/k**" option, PFE will not save changed mappings to the file you specify, or to the standard key map file, by default. When saving you will need always to give an explicit file name.

FROM THE KEY MAPPING DIALOG

To load a set of key mappings from another file when PFE is running, select the **Options Key Mapping** command, which starts the key mapping dialog. Clicking the **Load** button will cause PFE to ask you the name of the file of mappings that you want to load.

Changing The Active Prefix Keys

By default, no prefix keys will be active when you start PFE. This means that all the control keys are available to be mapped to functions, and you will not be able to use any two-character combinations to perform actions.

To activate or de-activate prefix keys, select the **Options Key Mapping** command, which starts the key mapping dialog, and click the **Configure** button to the right of the drop-down list of prefix keys.

This starts a sub-dialog that lets you activate or deactivate keys as prefixes. You can use any or all of **Esc** or **Ctrl+A** to **Ctrl+Z** as prefixes.

Keyboard Macros

PFE is able to record key strokes and menu selections in a *keyboard macro*, which you can replay as many times as you wish to make repetitive operations easier.

The **Macro Start Recorder** command will start the recorder. PFE will record anything you type into an edit window, and any menu items that you select into the keyboard macro buffer; recording will continue until you use the **Macro Stop Recorder** command.

Once you've recorded a keyboard macro, you can replay it with the **Macro Replay** command. PFE will re-issue all the keystrokes and menu selections that have been recorded, just as if you were performing them yourself.

It's up to you to ensure that it's sensible to replay a keyboard macro. For example, if you don't have a file open, it's not possible for PFE to insert any characters. So that you don't accidentally ask PFE to perform some impossible operation, replay of the keyboard macro will be automatically halted if any command or keystroke fails. This can be very useful if you're performing searches: macro replay will stop if the target string is not found.

Although it records menu selections made by both keyboard and mouse, PFE does *not* record other mouse operations. You should, therefore, take care to use only the keyboard equivalents of mouse operations when you're recording a macro.

PFE also does not record what you type into dialog boxes. Thus, if you record an **F2** key for a search, you'll see the dialog box every time you replay the key, and will have to fill in the details again. In many cases, this is how you would want the editor to operate, but sometimes you actually want to repeat a search operation exactly every time. To achieve this, you can type **F2** to perform a search *before* you start recording; then, within the macro you can use the **Shift+F2** key to repeat the search exactly without using the dialog box.

PFE provides only *one* keyboard macro; starting a recording will automatically delete any previous macro. In a future release you'll be able to select from multiple keyboard macros, and to record them to files for use in subsequent edit sessions.

Ending a PFE Session

You can end your PFE session by selecting the **Close** item in the system menu, by double clicking in the system menu box, or by using the **File Exit** command.

Before closing, PFE will check whether you have made any edit changes that have not yet been saved to disk. If you have, it will prompt you for each unsaved file in turn, asking you if you want to save the changes or to discard them, You also have the option of cancelling the process and remaining in PFE.

Exiting Windows

When you're developing programs that may not behave well and cause damage to the system, you'll often find it necessary to exit Windows and restart it, and maybe even to reboot your machine. PFE provides you an option on its *system menu* to close Windows in a variety of ways very quickly and simply.

The facilities provided by this option vary, depending on whether you're using Windows 3 or Windows NT.

Exiting Windows 3
Exiting Windows NT

Exiting Windows 3

To exit from your Windows 3 system, click on the system menu box at the top left corner of the main window and select the **Exit Windows** item. This starts a dialog that lets you perform on of the following actions:

- Exit from Windows 3 and return to your DOS prompt
- Exit from Windows 3 and then restart it automatically
- Exit from Windows 3 and then perform a warm reboot of your system
- Exit from Windows 3, run a single DOS command automatically, then restart Windows automatically

Exiting Windows NT

To exit from your Windows NT system, click on the system menu box at the top left corner of the main window and select the **Exit Windows** item. This starts a dialog that lets you perform on of the following actions:

- Shutown all the applications that you have started and log off from the system
- Shutdown all applications in the system, and close down Windows NT. You will be notified when it is safe to power your computer off
- Shutdown all applications in the system, close down Windows NT, and automatically reboot it

Running DOS Commands

PFE gives you two ways of interacting with DOS applications.

- You can start one or more DOS sessions running your command processor, within which you can run any DOS program. These sessions run independently of PFE; if you want to edit the output of any program you run in one, you'll need to use the clipboard to paste it into an edit window

- You can start a DOS application, such as a compiler, and arrange for PFE to capture its output as it runs. When the command completes, you'll see the results in an edit window. You can run external commands, or those internal to your command processor, in this way.

Starting A DOS Prompt
Running A Command And Capturing Output

Starting A DOS Prompt

The **Execute DOS Prompt** command starts up a DOS session, in which you can issue any DOS commands you wish.

If you have defined a command processor with a **comspec** statement in the **[options]** section of the initialisation file, this program will be run.

If not, PFE will look at the environment variable **COMSPEC**, and if this is defined, will run the program named in it.

If neither of the above options specifies a command processor, PFE will start **command.com** (for the Windows/16 version) or **cmd.exe** (for the Windows NT version)

The starting directory for the DOS session will be PFE's own current directory. This can be changed through the standard file opening dialogs run by items in the **File** menu.

You can start as many DOS sessions as you wish, and you can continue to use PFE while a session is running.

DOS sessions started in this way run independently of PFE. If you wish to edit anything that appears in their windows, use the standard clipboard mechanisms to paste it into PFE.

Running A Command And Capturing Output

PFE allows you to issue a DOS command, such as a compiler driver, and capture the output into a new edit window for inspection.

To do this, use the **Execute DOS Command To Window** command. This starts a dialog that lets you specify various things about the command you want to run

The dialog allows you to set:

- The command line to be executed. This can be any legal DOS command, including built-in commands such as **dir.** If you are using the **4DOS** command processor, you can also use aliases and command sequences separated by the current delimiter character. You can also have PFE substitute parts of the name of the file showing in the current window automatically for you.

- The working directory to be used. PFE will make this the current directory for the command process before the command is issued (PFE's own current directory setting is not affected)

- Whether PFE is to minimize itself into an icon once the command has been issued

- Whether PFE is to sound a beep once the DOS command has completed.

- Whether PFE is automatically to save any altered files before running the command

- Whether PFE is to show the output in any previously used command output window, or to create a new one

- Whether the output window is to scroll automatically to show you the end of the output, or is to show you the beginning

The values you set in the dialog box are remembered throughout the editing session, enabling you quickly to re-issue the same command with the minimum of effort.

You can run only one DOS command at a time in this way.

In order to run a DOS command, PFE requires you to have a helper module in the same directory as the main executable. The helper module is responsible for actually running the DOS command and redirecting its output.

Under Windows NT, the helper module is **$pfeds32.exe**. Under Windows/16, the module is called **$pfedos.exe**, and has an associated PIF file **$pfedos.pif**. As supplied, this PIF file runs the command in full-screen and exclusive modes; if you are working in 386 enhanced mode, you may wish to use the PIF editor to modify this.

The **Execute Repeat DOS Command To Window** command allows you to repeat the last command you issued with this dialog without needing to specify the details again. Note that if the command line you gave contained substitution points, these will be re-evaluated.

This functionality is not available if you are running the Windows/16 version of PFE in the Win16 subsystem of Windows NT.

Substituting The Current File Name In Commands

Substituting The Current File Name In Commands

PFE gives you a quick way to include the name of the file showing in the current window in the command lines specified in the dialogs you start with the **Execute DOS Command To Window** and **Execute Launch Application** commands.

You mark a place in the command line where part of the full path name is to be substituted with a "**%**" character, followed by a letter to show which part you require. You can specify any of these substitutions:

%d	The directory part of the path name, without a trailing "****"
%e	The extension part of the path name, without a leading "**.**"
%f	The filename and extension parts of the path name, separated by a "**.**"
%n	The filename part of the path name
%p	The entire path name
%u	The drive part of the path name, followed by a "**:**"
%%	A single "**%**" character

For example, suppose that the file showing in the current window was named **`c:\software\graphit\source\main.c`**

The command line

```
nmake %n.obj
```

would be expanded by PFE to be

```
nmake main.obj
```

and the command line

```
dir %u%d\*.c
```

would expand into

```
dir c:\software\graphit\source\*.c
```

If you don't have a file open, or if the current window does not have a file name associated with it - for instance, if it is a window showing command output, or was created with a **File New** command and you have not yet associated a file name with it - substitution will fail and you will see an error message.

The expanded command line must also not be longer than 512 bytes in total size. Note that this allows you to generate command lines that are longer than will be accepted by either the operating system or the command processor program that is asked to run them.

Launching Windows Applications

The **Execute Launch Application** command gives you a convenient way of starting another Windows or DOS application running. If you're developing a program, you'll find it a useful for starting up the latest version quickly.

Applications started in this way run independently of PFE. To run a DOS application such as a compiler, and see its output in an edit window, see the section on Running A Command And Capturing Output.

The command starts a dialog that lets you specify various things about the application you want to run. You can set:

- The command line to be executed. You can have PFE substitute parts of the name of the file showing in the current window automatically for you if you wish.

- The working directory to be used. PFE will make this the current directory for the command process before the command is issued (PFE's own current directory setting is not affected)

- Whether PFE is to minimize itself into an icon once the command has been issued

- Whether PFE is automatically to save any altered files before starting the task

The values you set in the dialog box are remembered throughout the editing session, enabling you quickly to start the same application again with the minimum of effort.

If you check the **Save Changed Files** box, PFE will check whether any of the files that you are editing have been altered, and will ask you if you want to save them. You should take care if you reply **Yes** to the question, as *all* the files that have been altered will be saved. Note that windows holding un-named files are not checked for changes.

You can launch as many independent applications as you wish.

Windows Tools

PFE allows you easily to launch Windows tools, such as the Dialog Editor, the Image Editor, and the CodeView debugger, by name from a list that you can configure.

When you add a Windows tool to the list you assign it a *tool name*, which can be any string up to 32 characters long. You associate with this name the command string you want to execute and the initial directory; then, when you want to start the tool, simply pick the name from the list.

You can have as many tool names referring to an application as you want: you could thus have a set that each invoke an application with slightly different command-line arguments.

Adding A Windows Tool Entry

To add an entry to the list of Windows Tools, you need to use the **Execute Configure Windows Tools** command. This starts a dialog that lets you manipulate the list of tools.

For each tool name, you can specify:

- The command line to be executed
- Whether PFE is to make itself into an icon before launching the tool
- Whether is PFE is automatically to save any changed files before launching the application

The Execute Configure Windows Tools Dialog

Changing A Windows Tool Entry

To change an entry in the list of Windows Tools, you need to use the **Execute Configure Windows Tools** command. This starts a dialog that lets you manipulate the list of tools.

For each tool name, you can specify:

- The command line to be executed
- Whether PFE is to make itself into an icon before launching the tool
- Whether is PFE is automatically to save any changed files before launching the application

Deleting A Windows Tool Entry

To delete an entry from the list of Windows Tools, you need to use the **Execute Configure Windows Tools** command. This starts a dialog that lets you manipulate the list of tools.

Launching A Windows Tool

To launch a tool that you've defined in the list of Windows tools, use the **Execute Launch Windows Tool** command. This starts a dialog that lets you select one of the tools you've configured.

You can launch the tool with the details you set when you configured it with the **Execute Configure Windows Tool** command, or you can change any details you wish.

PFE records the details that you set, and uses these as the default when you launch this tool again; however, you can return to the configured settings with a single button click.

The application that you launch runs independently of PFE.

Controlling PFE Over A DDE Link

Unless you started it with the command line option "**/m**", or had the run-modes value in the **[options]** section of the initialisation file set to 1, PFE will act by default as a **DDE Server**, allowing you to control its actions from some other program. This program will need to have the ability to act as a **DDE Client**; many standard applications already support this.

If you are using an existing Windows application as a DDE client you will need to consult its documentation for how to perform the necessary operations. You can easily write your own client application in Visual Basic (the DDETester application supplied with PFE demonstrates many of the techniques you will need); or you can write at API level using the Microsoft **DDEML** support library described in the SDK documentation and third party books.

Depending on how many DDE commands your application needs to send to PFE, and how frequently, it may be adequate for it to operate by establishing a DDE connection, issuing the command, and then closing the connection.

If, though, the application is going to make heavy use of DDE, it will be more efficient for it to establish a conversation once and leave it established until it has finished. If PFE detects that there are DDE conversations in progress when the user tries to exit from it, it will show a dialog warning that a client application is still operating, and asking whether the user wishes to abandon the closedown.

In this beta release of PFE, the DDE interfaces are still under development, and there may be considerable changes from what is described here in the final released version.

Starting And Stopping A DDE Server

Normally, when you start PFE, it will run as a DDE server automatically. However, if you start PFE running in multiple-instance mode, either with the "**/m**" command line option, or by having set the **run-modes** value to 1 in the **[options]** section of the initialisation file, it will not automatically act as a server.

If you do want an instance of PFE that you started in this way to act as a DDE server, you can use the **Options DDE Server** command to activate server mode. You can also use this command to turn the server off; the menu item will have a check mark beside it when the server is running.

Normally, you should try to ensure that only one instance of PFE is running as a DDE server at any time. Although it is permissible to run multiple servers, you can't then predict which server an application will connect to at any time. There is thus the real chance that an application may send DDE commands to the wrong PFE instance.

Issuing Commands Over A DDE Link

DDE commands are sent to PFE using the standard DDE methods. Your application should open a DDE link to PFE specifying the relevant DDE service name, and a topic name of either "**Editor**" or "**System**". From Visual Basic for example, you would perform a *Link.Execute* operation. The DDE item name is not used.

Once the link is established, you can send one or more command strings to be executed. The command string should have the general form

```
[CommandName(argument,...)]
```

where the arguments you supply within the parentheses will vary depending on the command. The command name can be written in any mixture of upper- and lower-case letters. A maximum of 128 bytes can be sent in any transaction.

PFE will return a response of **DDE_FACK** immediately if the command syntax is ostensibly correct, or **DDE_FNOTPROCESSED** if there is some obvious error. The command itself is executed asynchronously to the DDE client, which will regain control immediately after sending it. If the last DDE command resulted in a lengthy action - such as loading a large file, or asking the user for input in a dialog box - the server may not be able to accept a new command immediately. In this case it will return a reply **DDE_FBUSY** to the client, which should wait for a suitable interval before attempting the command again.

To determine whether any DDE link command worked or not you need to request the data for the item "**Result**" from the "**Editor**" topic. This data item will contain the string "**OK**" if the command succeeded, "**Error**" if it failed, and "**Busy**" if the server has not yet completed its processing.

List Of DDE Commands

Requesting Data Over A DDE Link

The PFE DDE server allows you to request items of data from your application. For example, you can ask PFE whether the DDE server is busy, or what the result of the last DDE command was; you can also find out about the state of files that are being edited and so on.

To request data, you need to open a DDE link to the relevant DDE service name with an appropriate topic. The link item is the name of the information that you require. From Visual Basic, for example, you would perform a *Link.Request* operation. The reply will always be a string of characters: numbers are converted into ASCII digits.

List Of DDE Data Items

Pasting Data Over A DDE Link

PFE gives you several ways of pasting data into the current file over a DDE link.

Using the EditInsert() command

For small amounts of data, you can send the DDE command **EditInsert()**, giving the string to be pasted as the argument. This will often be the most convenient technique to use.

Using the clipboard

With this method, you should first place the data to be pasted into the Windows clipboard in **CF_TEXT** format, with the end of each line marked by a CR-LF bytes pair and a null byte marking the end of data.

Then, you should issue the DDE command **EditPaste()**.

The command will run asynchronously to the client application. To determine whether it succeeded or failed, you will need to request the data item "**Result**" from the "**Editor**" topic.

Using DDE Poke

With this method, you initiate a conversation with the PFE DDE server using the relevant DDE service name, the topic "**Editor**" and item "**Paste**".

Your application should first place the data to be pasted into a shareable global memory segment in **CF_TEXT** clipboard format, where each line is ended by a CR-LF byte pair, and a final null byte marks the end of the data.

It should then perform a DDE **Poke** operation passing the handle to the memory segment. From Visual Basic, for example, you would execute a *Link.Poke* command.

PFE will reply **DDE_FACK** if the command was accepted, **DDE_FNOTPROCESSED** if there was an error, and **DDE_FBUSY** if the server was busy and could not accept the command.

Under Windows 3, you may not paste more than 64 kilobytes of data in a single operation. Under Windows NT there is no effective limit.

The paste operation will normally run asynchronously to the client application. To determine whether it succeeded or failed, you will need to request the data item "**Result**" from the "**Editor**" topic.

Opening Files Over A DDE Link

PFE gives you two distinct ways of opening files across a DDE link.

Using DDE Commands

The DDE commands **FileOpen()** and **FileView()** each allow you to open a single file, for editing or in read-only mode respectively. For opening small numbers of files, it will be most convenient to use this method.

The file names you give in these commands may include the normal DOS wildcard characters.

Using DDE Poke

To open a larger number of files, you can perform a DDE Poke operation, passing all the names in a single block of memory. To do this, you should

- Create a suitably large block of shareable global memory (under Windows 3.1, this block must not exceed 64 kilobytes in size)
- Place the name of the directory to be used by default with each file name at the start of the block, followed by a single space.
- Place the names of all the files following this, separating the names from each other by a single space. Files that are in the default directory should be stored without a directory path.
- Terminate the block with a single null byte.
- Perform a DDE Poke operation using the topic "**Editor**" and item name "**OpenFiles**" to open the files for editing, or the item name "**ViewFiles**" to open them in *read-only* mode.

PFE will open each file in turn for editing. If a file name does not start with a drive specification or a "\" character, the name of the default directory specified at the start of the block will be prepended to it; otherwise, the file name will be used exactly as presented.

File names may contain the normal DOS wildcard characters.

For example, a data block that contained the string

```
c:\source main.c c:\windows\win.ini ..\std.h *.h
```

would open the files **c:\source\main.c**, **c:\windows\win.ini**, **c:\source\..\std.h** and all the files matching the wildcard pattern **c:\source*.h**

Reference Section

This section of the help file contains reference material on PFE.

Command Line Options

When you start PFE, you can give several options on the command line to control its actions. These options should follow the name of the PFE executable itself, and precede any file names.

RUN MODE OPTIONS

These options control PFE's *run mode*, or whether more than one instance of it will run at a time. You can set the default run mode by setting the **`run-mode`** value in the **[options]** section of the initialisation file, but you can override that value when you start PFE with a command line option.

You can give one of the following options to control the run mode, overriding the initialisation file:

/m If the option is given, the command will start a new instance of PFE whether any previous instance is running or not. If it is absent, PFE will attempt to pass control to any other instance that is running as a DDE server, and will start as a separate instance only if this fails.

An instance of PFE started with this option present will not act as a DDE server unless the "**/d**" option was also used.

/s If this option is given, the command will look for any previously-running instance, and will activate it instead of running a new one. If you've given any files on the command line, at least one running instance must also be acting as a DDE server.

If no instance is already running, or if it can't be activated, a new instance will start.

CONFIGURATION OPTIONS

These options affect how PFE configures itself when it starts up:

/d If this option is given, an instance of PFE started with the "**/m**" option will act by default as a DDE server; without it the instance will not have the server enabled when it starts.

/k pathname

This option specifies a key map file to be loaded on start-up instead of the **pfe.key** file in your Windows directory. **Pathname** can specify any valid key map file; if it is the single character "**-**" (minus) PFE will not load any key mapping at all, but will use the built-in default values described in this help file.

Using this option implies the "**/m**" option; a new instance of PFE will be started whether or not any others are running, and will not by default run as a DDE server.

Whatever value you give for **pathname**, using this option will disable *all* use of the default **pfe.key** file. The **Save** button in the key mapping dialog will not be available when you change the mapping. To save to this file, you will need to use the **Save As** button and give the name explicitly.

OTHER OPTIONS

/v If this option is given, PFE will open any files that you name on the command line in *read-only* mode.

Examples

A command line like this

```
pfe data.txt results.txt
```

will start PFE, or activate a previously-running instance that was acting as a DDE server, and open the files **data.txt** and **results.txt** in the current directory.

The line

```
pfe /v data.txt results.txt
```

would act in the same way; but the files would be opened in read-only mode.

The command line

```
pfe /m
```

will start a new instance of PFE, whether or not any others were also running. The DDE server will not be enabled when the instance starts.

The command line

```
pfe /k mapping.key
```

will start a new instance of PFE and load the key mappings contained in the file **mapping.key** in the current directory. The DDE server will not be enabled when the instance starts.

Dictionary Of Functions

PFE allows you to map keyboard keys or key sequences to the following functions, which appear in the functions list of the key mapping dialog.

Not all the functions that you can map to keys have equivalent menu commands; those that do have **[*]** shown following the description.

CaretBottomOfWindow	Moves the caret to the start of the lowest line visible in the window
CaretBottomOfWindowSelect	Moves the caret to the start of the lowest line visible in the window, extending the selection
CaretDown	Moves the caret one line down
CaretDownSelect	Moves the caret one line down, extending the selection
CaretEndOfLine	Moves the caret to the end of the current line
CaretEndOfLineSelect	Moves the caret to the end of the current line, extending the selection
CaretEndOfFile	Moves the caret to the end of the current file
CaretEndOfFileSelect	Moves the caret to the end of the current file, extending the selection
CaretLeft	Moves the caret one character to the left
CaretLeftSelect	Moves the caret one character to the left, extending the selection
CaretLeftWord	Moves the caret one word to the left
CaretLeftWordSelect	Moves the caret one word to the left, extending the selection
CaretPageDown	Moves one page down in the current file
CaretPageUp	Moves one page up in the current file
CaretRight	Moves the caret one character to the right
CaretRightSelect	Moves the caret one character to the right, extending the selection
CaretRightWord	Moves the caret one word to the right
CaretRightWordSelect	Moves the caret one word to the right, extending the selection
CaretStartOfFile	Moves the caret to the start of the current file
CaretStartOfFileSelect	Moves the caret to the start of the current file, extending the selection
CaretStartOfLine	Moves the caret to the start of the current line
CaretStartOfLineSelect	Moves the caret to the start of the current line, extending the selection
CaretStartOfText	Moves the caret to the first non-white-space character in the current line

CaretStartOfTextSelect	Moves the caret to the first non-white-space character in the current line, extending the selection
CaretTopOfWindow	Moves the caret to the start of the top line in the window
CaretTopOfWindowSelect	Moves the caret to the start of the top line in the window, extending the selection
CaretUp	Moves the caret one line up
CaretUpSelect	Moves the caret one line up, extending the selection
EditCancelSelection	Removes any highlight from selected text in the current window
EditClearUndo	Forgets the details of all edit actions performed up to this point on the current file, freeing off the memory used to hold them **[*]**
EditCopy	Copies the selected text to the clipboard **[*]**
EditCut	Copies the selected text to the clipboard and deletes it **[*]**
EditDeleteSelection	Deletes the selected text
EditDeleteBackwards	Deletes the character to the left of the caret
EditDeleteForwards	Deletes the character to the right of the caret
EditDeleteLine	Deletes the whole of the line that the caret is in **[*]**
EditDeleteToEndOfLine	Deletes from the position of the caret to the end of the current line **[*]**
EditGotoLine	Starts a dialog that prompts for the line number to go to in the current file **[*]**
EditInsertHardTab	Inserts a TAB character, whether or not the current window's Window Mode is set for soft tabbing
EditInsertSoftTab	Inserts the required number of spaces to bring the caret to the next TAB stop position, even if the current window's Window Mode is not set for soft tabbing
EditInsertTab	Inserts either a TAB character or the required number of spaces to bring the caret to the next TAB stop position, depending on whether the current window's Window Mode is set for hard or soft tabbing
EditMarkUnchanged	Removes PFE's indication that the current file has been altered
EditNewLine	Inserts a new line character at the position of the caret
EditPaste	Pastes the contents of the clipboard into the current file at the position of the caret **[*]**
EditSearch	Searches for a string in the current file **[*]**
EditSelectLine	Selects all the characters in the line that the caret is within, moving the

	caret to the start of line and scrolling the window, if necessary, to make it visible
EditSelectWord	Selects the whole word that the caret is currently within **[*]**
EditShowCaret	Adjusts the position of the text in the current window so that the caret is as close to the central line as possible **[*]**
EditShowNextLine	Scrolls the window upwards to show the next line in the file. The caret remains in the same place within the file
EditShowNextPage	Scrolls the window upwards to show the next page in the file. The caret remains in the same place within the file
EditShowPreviousLine	Scrolls the window downwards to show the previous line in the file. The caret remains in the same place within the file
EditShowPreviousPage	Scrolls the window downwards to show the previous page in the file. The caret remains in the same place within the file
EditSplitLine	Splits the current line at the position of the caret, leaving the caret unmoved
EditRepeatLastFind	Repeats the last find operation **[*]**
EditRepeatLastReplace	Repeats the last replace operation **[*]**
EditReplace	Replaces occurrences of one string with another string **[*]**
EditTextIndent	Indents the current line, or all the lines in a selection, by one tab stop **[*]**
EditTextInsertASCIICode	Starts a dialog to insert a character specified by its ASCII code **[*]**
EditTextLowercaseSelection	Converts all the letters in the current selection to lower case **[*]**
EditTextMatchBrace	Moves the caret to the brace character matching the one currently under it, having regard to language syntax **[*]**
EditTextMatchBraceSelect	Moves the caret to the brace character matching the one currently under it, having regard to language syntax, and highlights all the text between and including the braces **[*]**
EditTextTransposeCharacters	Exchanges the character under the caret with the one to its left **[*]**
EditTextUndent	Undents the current line, or all the lines in a selection, by one tab stop **[*]**
EditTextUppercaseSelection	Changes all the text that is currently selected to be upper case **[*]**
EditTextWidenBraceSelect	Highlights all the text between and including the first matching pair of brace characters that includes either the currently-highlighted text, or the character under the caret if no text is highlighted **[*]**

EditUndo	Undoes the last edit action you performed **[*]**
ExecConfigureTools	Starts a dialog to configure details of program development tools **[*]**
ExecControlPanel	Starts the Windows Control Panel **[*]**
ExecDOSCommand	Runs a DOS command, capturing output in an edit window **[*]**
ExecDOSPrompt	Runs a DOS command-line shell **[*]**
ExecLaunchApp	Starts a Windows or DOS application to run independently of PFE **[*]**
ExecLaunchTool	Starts a program development tool **[*]**
ExecFileManager	Runs the Windows File Manager **[*]**
ExecPrintManager	Runs the Windows Print Manager **[*]**
ExecRepeatDOSCommand	Repeats the last DOS command run with the **ExecDOSCommand** function **[*]**
FileAbandon	Closes the file being edited in the current window, discarding any unsaved changes
FileClose	Closes the file being edited in the current window, prompting if there are unsaved changes **[*]**
FileCloseAll	Closes all the files being edited, prompting if any of them contain unsaved changes **[*]**
FileExit	Terminates the PFE editing session **[*]**
FileInsert	Inserts a file into the current one at the position of the caret **[*]**
FileMail	Mails the current file **[*]**
FileName	Changes the name of the file associated with the current window **[*]**
FileNew	Creates a new empty edit window for an unnamed file **[*]**
FileOpen	Opens an existing file for editing **[*]**
FilePrint	Prints the current file **[*]**
FilePrintSetup	Runs a dialog that allows printer information to be configured **[*]**
FileSave	Saves the current file to disk, provided that it has been altered **[*]**
FileSaveAs	Saves the current file to disk, allowing the output file name to be specified **[*]**
FileSaveAll	Saves all altered files and templates **[*]**
FileView	Opens an existing file in *read-only* mode **[*]**

FileWrite	Writes a file to disk whether or not it has been changed, allowing you to specify the name of the output file **[*]**
HelpAbout	Shows PFE's *About* box **[*]**
HelpCommands	Displays the PFE Help File at the **Commands** topic **[*]**
HelpContents	Displays the PFE Help File at the **Contents** topic **[*]**
HelpContextHelp	Starts the Windows help engine to give help on either the highlighted text in the current window, or the word under the caret **[*]**
HelpOnHelp	Starts the Windows help engine to give help about itself **[*]**
HelpProcedures	Displays the PFE Help File at the **Procedures** topic **[*]**
HelpReference	Displays the PFE Help File at the **Reference** topic **[*]**
HelpUserDefined1	Starts the Windows help engine, using the first help file configured in the **[help-files]** section of the initialisation file **[*]**
HelpUserDefined2	Starts the Windows help engine, using the second help file configured in the **[help-files]** section of the initialisation file **[*]**
HelpUserDefined3	Starts the Windows help engine, using the third help file configured in the **[help-files]** section of the initialisation file **[*]**
HelpUserDefined4	Starts the Windows help engine, using the fourth help file configured in the **[help-files]** section of the initialisation file **[*]**
HelpUserDefined5	Starts the Windows help engine, using the fifth help file configured in the **[help-files]** section of the initialisation file **[*]**
MacroReplay	Replays the last recorded keyboard macro **[*]**
MacroStartRecorder	Starts recording keystrokes and menu selections **[*]**
MacroStopRecorder	Stops recording keystrokes and menu selections **[*]**
OptionsCurrent	Starts a dialog to change the current window and file modes **[*]**
OptionsDDEStart	Starts the DDE server
OptionsDDEServer	Stops the DDE server
OptionsDDEToggle	Starts the DDE server if it is not running, and stops it if it is **[*]**
OptionsDefault	Starts a dialog to change the window modes for specific file types **[*]**
OptionsKeyMappings	Starts a dialog to change the mapping of keys to functions **[*]**
OptionsPrefixKeys	Starts a dialog that allows the user to define what prefix keys are active
OptionsResetModes	Sets the window modes for the current window, and the file modes for the file that it is showing, to those appropriate for the file's type **[*]**

OptionsScreenFontAnsi	Sets the screen font to be the standard non-proportional ANSI font **[*]**
OptionsScreenFontOEM	Sets the screen font to be the standard OEM font **[*]**
OptionsScreenFontOther	Starts a dialog that allows the user to select any screen font **[*]**
OptionsScreenFontSystem	Sets the screen font to be the standard non-proportional system font **[*]**
OptionsToggleInsertMode	Switches between *insert* and *overwrite* modes
OptionsToggleLineNumbers	Turns line numbering on or off in the current window
OptionsToggleStatusBar	Makes the status bar visible or invisible **[*]**
OptionsToolbarBottom	Positions the tool bar at the bottom of the screen **[*]**
OptionsToolbarFloat	Makes the tool bar into a detached floating window **[*]**
OptionsToolbarHide	Makes the tool bar invisible **[*]**
OptionsToolbarLeft	Positions the tool bar at the left of the screen **[*]**
OptionsToolbarRight	Positions the tool bar at the right of the screen **[*]**
OptionsToolbarShow	Makes an invisible tool bar visible at its previous location **[*]**
OptionsToolbarTop	Positions the tool bar at the top of the screen **[*]**
SysSetMenuMode	Highlights the leftmost menu item on the main window's menu bar, so you can easily access menu items from the keyboard. The effect is the same as pressing and releasing the **Alt** key.
TemplateFileAttach	Attaches a template file so the templates within it can be used **[*]**
TemplateFileCreate	Creates a new, empty template file **[*]**
TemplateFileDetach	Detaches an attached template file **[*]**
TemplateFileSave	Saves an altered template file to disk **[*]**
TemplateDelete	Deletes a template from a template file **[*]**
TemplateEdit	Edits a template from within a template file **[*]**
TemplateFindMark	Searches in a forward direction from the position of the caret for the next occurrence of a *template marker* **[*]**
TemplateInsert	Inserts a template from a template file into the current file at the position of the caret **[*]**
TemplateInsertMark	Inserts a *template marker* into a template **[*]**
TemplateNew	Creates a new empty window suitable for editing a template **[*]**
TemplateStore	Stores a template in a template file **[*]**

TemplateStoreAs	Stores a template in a template file, allowing its name to be changed **[*]**
WindowArrangeIcons	Arranges all iconized windows neatly **[*]**
WindowCascade	Arranges all non-iconic windows in a cascade **[*]**
WindowClose	Closes the current window **[*]**
WindowDuplicate	Makes an exact duplicate of the current window **[*]**
WindowIconizeAll	Makes all existing windows iconic **[*]**
WindowMaximize	Maximizes the current window **[*]**
WindowNext	Switches control to the next non-iconic window **[*]**
WindowRestore	Restores the current window from iconized or maximized state
WindowTileHorizontal	Arranges all non-iconic windows in a tile pattern, maximizing their depth **[*]**
WindowTileVertical	Arranges all non-iconic windows in a tile pattern, maximizing their width **[*]**
WindowWiden	Adjusts the size and position of the current window to make it as wide as possible **[*]**

The Default Key Mappings

As supplied, PFE contains a set of key mappings that enable you to perform the most common operations quickly. You may use these as they stand, or may modify them to suit your preferences.

By default, no prefix keys are enabled, so that only single keys are available for mapping.

Description Of Default Key Mappings
Table Of Keys And Default Functions

Description Of Default Key Mappings

This section lists the keys that have default mappings, giving you a description of the action that they carry out and the equivalent menu command (if any).

For a table that lists the keys with the names of the functions used in the key mapping dialog, see the **Table Of Keys And Default Functions**.

FUNCTION KEYS

F1	Opens the PFE Help File at the **Contents** topic
Shift+F1	Executes the **Help Screen/Menu Help** command to enter interactive help mode
Ctrl+F1	Executes the **Help Context Help** command to give help on the highlighted text in the current window or the word under the caret
F2	Executes the **Edit Find** command to search for a string
Shift+F2	Executes the **Edit Repeat Last Find** command to repeat the last search operation
F3	Executes the **Edit Replace** command to replace strings
Shift+F3	Executes the **Edit Repeat Last Replace** command to repeat the last replace operation
F4	Executes the **Template Find Mark** command to search for a template mark
Shift+F4	Executes the **Window Tile Horizontal** command to arrange windows in a tile pattern
Alt+F4	Executes the **File Exit** command to end your PFE session
Shift+F5	Executes the **Window Cascade** command to arrange windows in a cascade pattern
Ctrl+F5	Executes the **Edit Show Caret** command to bring the line containing the caret to the centre of the window
F6	Executes the **Template Insert Mark** command to insert a template mark
F7	Executes the **Macro Replay** command to replay a keyboard macro
Shift+F7	Executes the **Macro Start Recorder** command to begin recording a keyboard macro
Ctrl+F7	Executes the **Macro Stop Recorder** to stop recording a keyboard macro
F9	Executes the **Template Insert** command to insert a template into the current window
F11	Executes the **Execute DOS Command To Window** command to run a DOS command and capture its output in a window
Shift+F11	Executes the **Execute Launch Application** command to launch a Windows application to run independently of PFE

Ctrl+F11	Executes the **Execute DOS Prompt** command to start a DOS session
F12	Executes the **Execute Launch Windows Tool** command to launch a tool from the list of configured Windows Tools
Shift+F12	Executes the **Execute Configure Windows Tools** command to let you configure the list of Windows Tools

MOVEMENT KEYS

For these keys, the default mapping is such that adding **Shift** to the combination causes a selection to be extended.

Note that where the **Alt** key is combined with either an arrow key, or one of **Home**, **End**, **PgDn**, **PgUp**, **Ins** or **Del**, you must use the keys in the extended key areas, and not those in the numeric keypad.

Up	Moves the caret up by one line
Shift+Up	Moves the caret up by one line, extending the selection
Down	Moves the caret down by one line
Shift+Down	Moves the caret down by one line, extending the selection
Left	Moves the caret left by one character
Shift+Left	Moves the caret left by one character, extending the selection
Ctrl+Left	Moves the caret left by one word
Ctrl+Shift+Left	Moves the caret left by one word, extending the selection
Right	Moves the caret right by one character
Shift+Right	Moves the caret right by one character, extending the selection
Ctrl+Right	Moves the caret right by one word
Ctrl+Shift+Right	Moves the caret right by one word, extending the selection
PgDn	Moves the caret down by one page
Shift+PgDn	Moves the caret down by one page, extending the selection
Ctrl+PgDn	Moves the caret to the start of the last line in the window
Ctrl+Shift+PgDn	Moves the caret to the start of the last line in the window, extending the selection
PgUp	Moves the caret up by one page
Shift+PgUp	Moves the caret up by one page, extending the selection
Ctrl+PgUp	Moves the caret to the start of the first line of the window

Ctrl+Shift+PgUp	Moves the caret to the start of the first line of the window, extending the selection
Home	Moves the caret to the start of the line
Shift+Home	Moves the caret to the start of the line, extending the selection
Ctrl+Home	Moves the caret to the start of the file
Ctrl+Shift+Home	Moves the caret to the start of the file, extending the selection
Alt+Home	Moves the caret to the first non-white-space character in the current line
Alt+Shift+Home	Moves the caret to the first non-white-space character in the current line, extending the selection
End	Moves the caret to the end of the line
Shift+End	Moves the caret to the end of the line, extending the selection
Ctrl+End	Moves the caret to the end of the file
Ctrl+Shift+End	Moves the caret to the end of the file, extending the selection

CONTROL KEYS

Ctrl+B	Executes the **Edit Text Match Brace** command to move the caret to the brace character matching the one currently under it
Ctrl+Shift+B	Executes the **Edit Text Match Brace Select** command to move the caret to the brace character matching the one currently under it and highlight all the text between and including the brace characters
Ctrl+C	Executes the **Edit Copy** command to copy selected text to the clipboard
Ctrl+G	Executes the **Edit Goto Line** command to move the caret to a specific line
Ctrl+H	Deletes the character to the left of the caret
Ctrl+I	Inserts a TAB character, or the required number of spaces to bring the caret to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft TABs.
Ctrl+K	Executes the **Edit Delete To End Of Line** command to delete everything from the caret position to the end of the line
Ctrl+Shift+K	Executes the **Edit Delete Line** command to delete the entire line that the caret is in
Ctrl+N	Executes the **File New** command to create a new, empty edit window
Ctrl+O	Executes the **File Open** command to open an existing file
Ctrl+Shift+O	Splits the current line at the position of the caret, leaving the caret unmoved

Ctrl+P	Executes the **File Print** command to print the file showing in the current window
Ctrl+Q	Executes the **Edit Text Insert ASCII Code** command to insert a character specified by its ASCII code number
Ctrl+S	Executes the **File Save** command to save the current file to disk
Ctrl+T	Executes the **Edit Text Transpose Characters** command to transpose the character under the caret with the one to its left
Ctrl+V	Executes the **Edit Paste** command to paste data from the clipboard into the current window
Ctrl+W	Executes the **Window Select** command to let you choose between many open windows
Ctrl+Shift+W	Executes the **Edit Text Widen Brace Select** command to highlight all the text between and including the closest pair of brace characters that encompasses either the currently-highlighted text, or the character under the caret if no text is selected
Ctrl+X	Executes the **Edit Cut** command to cut selected text to the clipboard
Ctrl+Z	Executes the **Edit Undo** command to undo the last edit action

OTHER KEYS

Enter	Inserts a newline
BackSpace	Deletes the character to the left of the caret
Del	Deletes the character to the right of the caret
Shift+Del	Executes the **Edit Cut** command to cut selected text to the clipboard
Ins	Toggles the current window between Insert and Overwrite modes
Shift+Ins	Executes the **Edit Paste** command to paste data from the clipboard into the current window
Ctrl+Ins	Executes the **Edit Copy** command to copy selected text to the clipboard
Keypad 5	Cancels the current selection, removing the highlight
Tab	Inserts a TAB character, or the required number of spaces to bring the caret to the next TAB stop, depending on whether the current window's Window Mode is set for hard or soft TABs.

Table of Keys And Default Functions

These tables show you the functions that are by default mapped to certain keys. For a description of the action of each function, consult the functions dictionary

A descriptive list of the action of each key is in the **Description Of Default Key Mappings** section.

FUNCTION KEYS

KEY	Used Alone	With Shift	With Ctrl	With Alt
F1	HelpContents	HelpScreenMenuHelp	HelpContextHelp	
F2	EditFind	EditRepeatLastFind		
F3	EditReplace	EditRepeatLastReplace		
F4	TemplateFindMark	WindowTileHorizontal		FileExit
F5		WindowCascade	EditShowCaret	
F6	TemplateInsertMark			
F7	MacroReplay	MacroStartRecorder	MacroStopRecorder	
F8				
F9	TemplateInsert			
F10	SysSetMenuMode			
F11	ExecDOSCommand	ExecLaunchApp	ExecDOSPrompt	
F12	ExecLaunchWindows-Tool	ExecConfigureWindows-Tools		

MOVEMENT KEYS

For these keys, the default mapping is such that adding **Shift** to the combination causes a selection to be extended.

Note that where keys are mapped in combinations including the **Alt** key, you must use the keys from the extended keyboard area, and not those in the numeric keypad

KEY	Used Alone	With Shift	With Ctrl	With Ctrl+Shift
Up	CaretUp	CaretUpSelect		
Down	CaretDown	CaretDownSelect		
Left	CaretLeft	CaretLeftSelect	CaretLeftWord	CaretLeftWordSelect
Right	CaretRight	CaretRightSelect	CaretRightWord	CaretRightWordSelect
PgUp	CaretPageUp	CaretPageUpSelect	CaretTopOfWindow	CaretTopOfWindowSelect
PgDn	CaretPageDown	CaretPageDownSelect	CaretBottomOfWindow	CaretBottomOfWindowSelect
Home	CaretStartOfLine	CaretStartOfLineSelect	CaretStartOfFile	CaretStartOfFileSelect
End	CaretEndOfLine	CaretEndOfLineSelect	CaretEndOfFile	CaretEndOfFileSelect

KEY	With Alt	With Alt+Shift
Home	CaretStartOfText	CaretStartOfTextSelect

CONTROL KEYS

KEY	Used Alone	With Shift
Ctrl+B	EditTextMatchBrace	EditTextMatchBraceSelect

Ctrl+C	EditCopy	
Ctrl+G	EditGotoLine	
Ctrl+H	EditDeleteBackwards	
Ctrl+I	EditInsertTab	
Ctrl+K	EditDeleteToEndOfLine	EditDeleteLine
Ctrl+N	FileNew	
Ctrl+O	FileOpen	EditSplitLine
Ctrl+P	FilePrint	
Ctrl+Q	EditTextInsertASCIICode	
Ctrl+S	FileSave	
Ctrl+T	EditTextTransposeCharacters	
Ctrl+V	EditPaste	
Ctrl+W	WindowSelect	EditTextWidenBraceSelect
Ctrl+X	EditCut	
Ctrl+Z	EditUndo	

OTHER KEYS

KEY	Used Alone	With Shift	With Ctrl	With Ctrl+Shift
Enter	EditNewline			
BackSpace	EditDeleteBackwards			
Del	EditDeleteForwards	EditCut		
Ins	OptionsToggleInsertMode	EditPaste	EditCopy	
Keypad 5	EditCancelSelection			
Tab	EditInsertTab			

PFE uses an initialisation file to record the values that should be carried over from one session to another; you can also place various customisation options in it.

As some options in the file may need to be different between the Windows 3 and Windows NT environment, and because both environments normally share the same Windows directory, different file names are used. In the Windows 3 environment, the initialisation file is called **pfe.ini**; under Windows NT it is **pfe32.ini**.

In both environments the file is held in your Windows directory.

Initialisation File Format

PFE uses an initialisation file to record the values that should be carried over from one session to another; you can also place various customisation options in it.

As some options in the file may need to be different between the Windows 3 and Windows NT environment, and because both environments normally share the same Windows directory, different file names are used. In the Windows 3 environment, the initialisation file is called **pfe.ini**; under Windows NT it is **pfe32.ini**

For both environments the file is held in your Windows directory, and can be edited with PFE or any other editor. However PFE will change many parts of the file when it exits. Unless specified, you should therefore edit sections with the Windows NotePad editor for safety.

The information contained in the file is in the standard format used by all Windows applications. The file is divided into *sections* by lines of the form

[options]

and all lines following this up to the next section name are treated as a group.

Within each section come lines of the general form

key string=argument,argument,....

The **key string** is text (which may include spaces) that PFE uses to locate a particular line; the comma-separated arguments that follow the **=** sign provide the data relevant to the key.

At this beta release the format of sections other than those given below is not finalised; the entire file will be documented for the first main release.

The [options] Section Of The Initialisation File

This section contains various items that allow you to customize how PFE behaves to suit your own preferences. PFE never alters the contents of this section itself.

You can safely use PFE itself to alter the contents of this section. Any changes you make will not take effect until you terminate PFE and restart it.

The lines that you can include are as follows:

allow-save-always=number

Specifies whether PFE should always allow you to save a file, whether or not it has been changed. If **number** is 0, the **File Save** command will be enabled only after you change the file; if **number** is 1, the menu option and tool bar button will always be available.

The default value is 0.

auto-file-action=number

Specifies what action PFE should take on starting up if no files are named on the command line. The values permitted for **number** are:

- **0** No action
- **1** Simulate a **FileNew** command and create an empty, unnamed window
- **2** Simulate a **FileOpen** command and show the dialog that allows you to choose which files to open

The default value is 0.

auto-format=number

Specifies whether PFE is to attempt automatically to detect that a file being opened is in UNIX format. If **number** is 0, PFE does not try to determine the format; if it is 1, it examines the terminator of the first line in the file, and if it is a single Line Feed character (LF) sets the file mode **Save in UNIX format** for the file.

The default value is 1.

backup-directory=name

Specifies the relative name of the sub-directory to hold backup copies of files when the **backup-mode** value is set to **0**. **Name** can be up to 8 characters long, and should contain only characters acceptable in a directory name. It may not contain path separators.

When taking a backup of file that is to be overwritten, PFE will create or use a sub-directory of the given name in the directory that contains the file itself.

If this option is not given, the directory name "**$PFEBK**" is used.

The value is not used if **backup-mode** is set to other than 1.

backup-mode=number

Specifies how PFE should take backups of files being saved. The values permitted for **number** are:

0 Backups are maintained in the same directories as the originals, in files that have the same name but a file type of "**.$$$**"

1 Backups are maintained in subdirectories of the directory holding the originals

5 No backups are maintained

The default value is 0.

comspec=pathname

Specifies the name of the command processor to be run by the **Execute DOS Prompt** command. The pathname should be an absolute one, including a drive specification.

If this option is absent, PFE will attempt to use the environment variable **COMSPEC** to locate the command processor. If this also is undefined, **command.com** (for the Windows/16 version) or **cmd.exe** (for the Windows NT version) will be started.

context-help-file=pathname

Specifies the name of the Windows help file to be used by the **Help Context Help** command or a double-click of the right mouse button over a word of text.

The pathname should either be absolute, or identify a Windows help file in a directory on the **PATH**.

If this option is absent, context help will not operate.

deselect-on-copy=number

If **number** is 0, PFE leaves the selected text highlighted after it is copied to the clipboard. If **number** is 1, the highlighting is removed.

The default value is 0.

dragdrop-flip=number

If **number** is 0, text drag-and-drop performs a *move* operation if no key is pressed, and a *copy* if the **Ctrl** key is down at the time of releasing the left mouse button. If **number** is 1 the actions are reversed, so that *copy* becomes the operation performed when no key is pressed.

The default value is 0.

max-undo-actions=number

Specifies how many undoable edit actions PFE will record for each file. You can set **number** to any value between 8 and 128; the default value is 32. Once the list becomes full, old actions are discarded to make room for new ones.

Depending on the nature of the edits you perform, maintaining the details of a large number of edit actions could consume a substantial amount of memory.

max-vertical-tile=number

If **number** is 0, PFE will leave room at the bottom of the screen to show one row of icons when tiling windows vertically. If it is 1, the entire depth of the screen is covered with the tiled windows.

The default value is 0.

minimize-on-empty=number

If **number** is 1, the main PFE window will become an icon when the final edit window in use is closed. If it is 0, the main window's state does not change.

The default value is 0

mru-files-shown=number

This option controls the maximum number of files from the most-recently-used files list that will be shown on the **File** menu; **number** can lie in the range 0 to 8. If the list contains more files than are displayed, an additional menu item **More Files** (or **Recent Files** if the value is zero) is added to the menu to allow them to be selected from a dialog.

The default value is 5; setting larger values may make the **File** menu too long for convenient display on a standard VGA display.

mru-list-size=number

This option controls the maximum number of file names that PFE will record in its list of most-recently-used files. **Number** can lie in the range from 0 to 64; setting it to 0 causes PFE not to maintain a list.

The default value is 5.

open-maximized=number

This option controls whether file windows are created in a maximized or a restored state. The values permitted for **number** are:

- **0** Create file window in a restored state
- **1** Create file window maximized always
- **2** Create file window maximized if the current file's window is also maximized
- **11** Create file window maximized if PFE's main window is maximized
- **12** Create file window maximized if PFE's main window is maximized and the current file's window is also maximized

Windows created to receive DOS command output, and those created by **Window Duplicate** command, are always created in a restored state whatever the value of this option.

The default value is 0.

run-mode=number

This options sets the default value for whether PFE will try to run as only one instance, or as multiple instances. The values permitted for **number** are:

0 Attempt to ensure that only one instance of PFE is running at any time

1 Always start a new instance of PFE

The default value is 0. The value set here can be overridden with the command line options **/m** and **/s**.

save-clears-undo=number

If **number** is 1, PFE will free the details of edit actions performed on a file when you save it to disk. This can result in a substantial saving in memory usage, but will mean that edit actions performed up to the time of saving can no longer be undone.

If **number** is 0, no change is made.

The default value is 1.

save-find-strings=number

If **number** is 1, the last eight strings that you have used in the Find or Replace dialogs will be recorded in PFE's initialisation file when you end your editing session. The strings will then be restored the next time you use PFE. If **number** is 0, no strings are recorded.

The default value is 0.

select-search-match=number

If **number** is 1, PFE will highlight a matching string at the end of a successful search operation. If it is 0, the text is not highlighted. The value may be over-ridden in the Find dialog box.

The default value is 1.

sound-beep=number

If **number** is 0, PFE will not generate any warning beeps when it displays message boxes indicating error conditions; if it is 1, certain messages will sound a beep to notify you.

The default value is 1. You cannot suppress the warning beeps that are the sole indication of an error (for example, when you try to insert a character in a read-only window)

start-maximized=number

If **number** is 1, PFE will always start with its main window maximized. If it is 0, PFE will start with the main window in a restored state, at the same size and position as it was at the end of the last session. If the option is absent, PFE will start either restored or maximized, depending on its state at the end of the last session.

toolbar-size=number

This options overrides the automatic sizing of the toolbar, which sets the buttons at a size appropriate for the screen resolution, and allows you to select the size required. The values allowed for **number** are

0 Size the toolbar automatically to suit the screen resolution

1 Use the small set of toolbar buttons, normally used on screens of 640x480 resolution

2 Use the medium set of toolbar buttons, normally used on screens of 800x600 resolution

3 Use the large set of toolbar buttons, normally used on screens of 1024x768 resolution

The default value is 0.

track-vertical-thumbtack=number

If **number** is 1, PFE will scroll the text in a window to follow the vertical scrollbar thumbtack when this is dragged with the mouse. If **number** is 0, the window is not updated until the left mouse button is released.

The default value is 1.

use-dragdrop=number

If **number** is 0, PFE will not perform drag-and-drop moving and copying of text. If it is 1, drag-and-drop is available.

The default value is 1.

The [fileopen-filters] Section Of The Initialisation File

This section allows you to specify the *filters* that appear in the common dialogs for opening and saving files. PFE never alters the contents of this section itself.

The filters appear in the list box at the lower left of the dialogs, and control what files are shown in the main list above them. For example, if you selected a filter "***.c;*.h**", only files with types "**.c**" and "**.h**" would be shown.

Within the **[fileopen-filters]** section each line represents one filter, and PFE inserts them into the list of filters in the order you give them. Each line has the form

text=filterlist

Where **text** is some descriptive text such as "**Source Files**" that you can choose yourself, and **filterlist** is a list of wildcard patterns, separated by commas.

For example, you might include the line

Source Files=*.c,*.h

to set up a filter to show only files with types of "**.c**" and "**.h**" .

PFE uses *only* the filters you specify, so normally you would want to include a line such as

All Files=*.*

probably as the last in the section to let you see files not covered by the other filters.

If you don't have a **[fileopen-filters]** section in your initialisation file, PFE uses a single filter of "***.***" to show all files.

You can safely use PFE itself to alter the contents of this section. Any changes you make will not take effect until you terminate PFE and restart it.

The [help-files] Section Of The Initialisation File

This section allows you to specify up to five items to be added to the **Help** menu. Each item causes PFE to invoke the Windows help engine on a specified help file.

Within the section, each entry has the format

```
menu_string=helpfile_path
```

Menu_string specifies what you want to see on the help menu for the item. If you include an "**&**" character in the string, the character following it will appear underlined, and will be used as the menu item's hot key.

Helpfile_path specifies the name of the Windows help file to be opened when you click the menu item. You must either give a full path name for the file, or it must be in a directory named in the **PATH** environment variable.

For example, a section like this:

```
[help-files]
Windows NT API=c:\win32hlp\win32wh.hlp
MSC 32-bit Compiler RTL=c:\win32hlp\msc.hlp
```

would add two items to the **Help Menu**.

You can safely use PFE itself to alter the contents of this section. Any changes you make will not take effect until you terminate PFE and restart it.

If you wish, you can map each of the added items to key strokes; the functions that correspond to them are **HelpUserDefined1** to **HelpUserDefined5**.

The [managers] Section Of The Initialisation File

This section allows you to specify the command lines to be executed by the **Execute File Manager**, **Execute Control Panel**, **Execute Print Manager**, **Execute Program Manager** and **Execute Task Manager** commands.

The entries defined for this section are:

control-panel=commandline

Specifies the command line run by the **Execute Control Panel** command

file-manager=commandline

Specifies the command line run by the **Execute File Manager** command

print-manager=commandline

Specifies the command line run by the **Execute Print Manager** command

program-manager=commandline

Specifies the command line run by the **Execute Program Manager** command

task-manager=commandline

Specifies the command line run by the **Execute Task Manager** command

You can safely use PFE itself to alter the contents of this section. Any changes you make will take immediate effect.

DDE Commands And Data Items

PFE can be controlled remotely by another application using a DDE link. PFE supports the standard DDE techniques, and can be used from any compliant application.

There are two ways of communicating with PFE. The application can send it **commands**, which cause PFE to take some action; or it request **data items**, which cause PFE to return it some information.

Controlling PFE Over A DDE Link

DDE Commands
DDE Data Items
DDE Poke Items

DDE Commands

The DDE commands currently supported are listed below. For details of how to use a DDE link to control PFE, see Controlling PFE Over A DDE Link

CaretBottomOfWindow(*select*)
Moves the caret to the start of the last line in the window, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretDown(*count,select*)
Moves the caret down by *count* lines. If *select* is 1 the characters it moves over are selected and shown in reverse video; if it is 0 they are not and any outstanding selection is cancelled.

CaretEndOfLine(*select*)
Moves the caret to the end of the current line, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretEndOfFile(*select*) Moves the caret to the end of the file, possibly extending a selection.

CaretHome(*select*) Moves the caret to the start of the current line, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretLeft(*count,select*)
Moves the caret left by *count* characters, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretLeftWord(*count,select*)
Moves the caret left by *count* words, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretRight(*count,select*)
Moves the caret right by *count* characters, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretRightWord(*count,select*)
Moves the caret right by *count* words, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretStartOfFile(*select*)
Moves the caret to the start of the file, possibly extending a selection.

CaretTopOfWindow(*select*)
Moves the caret to the start of the first line in the window, extending the selection if *select* is 1, and not extending it if *select* is 0.

CaretUp(*count,select*)
Moves the caret up by *count* lines, extending the selection if *select* is 1, and not extending it if *select* is 0.

EditCopy() Copies the currently selected text to the clipboard.

EditCut() Deletes the currently selected text, copying it to the clipboard.

EditDeleteBackwards(*count*)
Deletes the *count* characters to the left of the caret.

EditDeleteForwards(*count*)

Deletes the *count* characters to the right of the caret.

EditGotoLine(*number,select*)

Moves the caret to the start of the line specified by *number*. If the string is preceded by either a "**+**" or a "**-**" sign, PFE will move to the start of the line that is *number* after or before the current line , respectively. The string "**end**" may be used to represent the start of the last line in the current file. If *select* is "**1**", the text from the starting position to the start of the target line will be highlighted.

EditFind(*"string",options*)

Searches the current window for the specified *string*. Within *string*, special characters are represented with the same notation as used in the dialog started by the **Edit Find** command.

The *options* parameter is a numeric value that specifies how the search operation is to be carried out, made up by summing any values from this list:

1	Search in the reverse direction
2	Search is to be case sensitive
4	Highlight the matching string
8	Highlight all text from the starting position of the caret to the end of the matching string

The **Result** data item will return "**OK**" if the search string was found, and "**Error**" if it was not.

EditInsert("*string*")

Inserts the quoted string into the current file at the position of the caret. Within the string, the characters "**\f**" represent a Form Feed, "**\n**" represent a line break and "**\t**" a tab character. To insert a "\" character, specify it as "\\", to insert a quote character, specify it as two consecutive quote characters. The notation "**xnn**", where "**nn**" is two hexadecimal digits, allows you to specify an arbitrary character code (the null value "**\x00**" is not permitted).

EditPaste()

Causes the contents of the Windows clipboard (provided that it is in **CF_TEXT** format) to be pasted into the current file at the position of the caret. You can also use a DDE Poke operation to paste data.

EditSelectLine()

Selects all the characters in the line containing the caret. The caret is moved to the start of the line and the window is scrolled if necessary to make this visible.

FileAbandon()

Closes the current file. If the file has been altered, unsaved changes will be discarded.

FileClose()

Closes the current file. If the file has been altered, the user will see a dialog box asking him if he wishes to save the data.

FileInsert(*"filename"*)

Inserts the specified file into the current file at the position of the caret. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory. If the filename is omitted, PFE will show a dialog box to ask the user to supply it.

FileName(*"filename"*) Changes the name of the file associated with the current window. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory. This command will fail if the current window contains a template.

FileNew() Creates a new window for an un-named file.

FileOpen(*"filename"*) Opens the specified file for editing. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory. You can open several files at once with a DDE Poke operation using the item "**OpenFiles**".

The filename you specify in this command may contain the normal DOS wildcard characters.

FileSave() Writes the current file to disk, providing it has been changed since it was last saved. If the file is currently un-named, the user will see a dialog asking him for the file name it is to be saved under.

FileSaveAs(*"filename"*) Writes the current file to disk, changing its name in the process. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory.

FileView(*"filename"*) Opens the specified file in read-only mode. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory.

The filename you specify in this command may contain the normal DOS wildcard characters.

FileWrite(*"filename"*) Writes the current file to disk with the specified filename, whether or not it has been changed. The name by which the file is known to PFE, as shown in the window caption, is not altered. If the filename is not an absolute pathname, it is treated as being relative to PFE's current working directory.

WindowActivate(*id*) Activates the window with the specified *id* value. The *id* is a numeric string, returned by querying the data item **WindowID** when the desired window is current.

DDE Data Items

The supported items that you can request data from over a DDE link are listed below. For details of how to use a DDE link to control PFE, see Controlling PFE Over A DDE Link

Topic "Editor"

ColumnNumber	Gives you the number of the text column containing the caret in the current file, or "**Error**" if there is no file open. The leftmost column is numbered 1.
FileChanged	Tells you whether the current file has been altered since it was last saved. The reply will be the string "**Yes**" if the file has changed; and "**No**" if either the file has not changed, or there is no current file.
FileName	If the current window contains a file that has an associated file name, the reply will be that name. If there is no current window, or the window has no associated file name, it will be the string "**Error**".
FileWritable	Tells you whether the current file can be written or not (*i.e.* that *read-only* mode has not been set). The reply will be "**Yes**" or "**No**", or "**Error**" if no file is open.
LineNumber	Gives you the number of the line in the current file containing the caret, or "**Error**" if there is no file open. Depending on the size of the file, the number may exceed the capacity of a 16-bit integer.
Result	Gives you the result of the last command executed over the DDE link. The reply will be one of the strings "**OK**" or "**Error**" if a result is available, and "**Busy**" if the server is currently processing a DDE link command. The results of commands executed from the keyboard, menu or toolbar cannot be read by this method.
Status	Gives you the current status of the DDE server. The reply will be one of the strings "**Ready**" and "**Busy**".
VersionString	Gives you the version identification of PFE as a string in the form "*x.yy.zzz*" as shown in the dialog started by the **Help About** command.
WindowID	Gives you the unique id number of the current window as a numeric string. This id value can be used to identify the window for the whole of its life; id values are never re-used in any given PFE session. If no window exists, the string "**Error**" is returned.
WindowTitle	Gives you the string that is the caption of the current window

Topic "System"

Formats	A list of the clipboard formats supported by PFE's DDE server. This will be the string "**TEXT**"
SysItems	A list of items supported by PFE under the "**System**" topic, separated by tab characters
Topics	A list of the topics supported by PFE, separated by tab characters

Status	The current status of the DDE server. The reply will be one of the strings "**Ready**" and "**Busy**".

DDE Poke Items

The DDE Poke items currently supported are listed below. For details of how to use a DDE link to control PFE, see Controlling PFE Over A DDE Link

All DDE Poke operations should use the relevant DDE service name and the topic "**Editor**". The data should be passed in a shareable global data segment.

OpenFiles

This topic allows you to open a number of files for editing in one operation. The data should contain the name of the default directory to be used for each file, separated by one space from a list of filenames, also separated by one space. A null byte should mark the end of the list.

If a file name in the list does not contain a directory path, the default directory string will be prepended to it; otherwise it is used exactly as supplied.

The file names given in the list may contain the normal DOS wildcard characters.

ViewFiles

This topic allows you to open a number of files in *read-only* mode in one operation. The details are as for the **OpenFiles** topic above.

Paste

This topic allows you to paste data into the current window at the position of the caret. The data should be in **CF_TEXT** clipboard format, with the end of each line marked by a CR-LF byte pair, and a final null byte terminating the block.

Under Windows 3, you may not paste more than 64 kilobytes of data in one operation.

DDE Poke operations are carried out asynchronously to the client application. To determine whether they succeeded or failed, the client must request the data from the item "**Result**" in the "**Editor**" topic.

List Of Window Modes

PFE associates a set of **Window Modes** with every edit window, that tell it how to present text to you. The modes can be set with the **Options Current File/Window Modes** and **Options Default File/Window Modes** commands.

The window modes that you can set are these:

Auto Indenting	This mode is either **on** or **off**. When it is on, PFE automatically indents new lines to match the indent of the one above.
Language	This mode is a string which defines what language awareness PFE should use for the window. Currently, you can set this only to "**C**"
Overwrite	This mode is either **on** or **off**. When on, PFE replaces the characters under the caret by the ones you type.
Page Headers	This mode is either **on** or **off**. When printing, PFE heads each print output page with a title showing the name of the file, the date and time, and the page number if the mode is on.
Soft Tabs	This mode is either **on** or **off**. When it is off, PFE inserts a single tab character into the file when you press the TAB key. When the mode is on, PFE inserts a suitable number of spaces to bring the caret to the next tab stop, as defined by the **Tab Size** mode.
Show Line Numbers	This mode is either **on** or **off**. When it is on, PFE displays the number of each line in the window.
Strip Trailing Spaces	This mode is either **on** or **off**. When it is on, PFE removes any trailing space or tab characters from a line when you press **Enter** at the end of it.
Tab Size	This mode is a numeric value that defines the width of a tab stop on the screen.
Wrap Column	This mode is a numeric value that defines the column at which automatic text wrapping is performed. By default, the mode has the value of 72.
Wrap Enabled	This mode is either **on** or **off**. When it is on, PFE will automatically wrap the line you're typing in as you move past the column set by the **Wrap Column** mode.
Wrap Long Lines	This mode is either **on** or **off**. When printing a file, PFE will fold lines that do not fit the page if the mode is on, and truncate them if it is off.

Language Awareness

PFE contains some awareness of the format of programming languages, to help you when writing program sources. Currently only the C language is supported.

To select language awareness, you need to specify the language in the window modes for the window you are editing in. You can specify that any file with a given file type should use a particular language automatically, or you can change the current window's language at any time.

When editing files that have language type set to "**C**", PFE provides these facilities:

- A **#** character typed in an otherwise empty line is always moved to column 1

- A closing **}** brace typed in an otherwise empty line is moved to the same column as the matching preceding opening brace **{**, providing that this is the only character in its line. In locating the opening brace, characters within string delimiters, comment delimiters or in pre-processor directives will be ignored.

The **Edit Text Match Brace**, **Edit Text Match Brace Select** and **Edit Text Widen Brace Select** commands are also set to operate using opening braces from the set **{ [** and **(**, and closing braces from the set **}]** and **)**.

The Options Default File/Window Modes Dialog

This dialog allows you to set the file and window modes that PFE is to apply when you open existing files or create new ones.

Steps

1. Specify what sort of file you're setting modes for.

   To set default modes for files created with the **File New** command, click the **New Files** button

   To set default modes for files of a given file type, that will be applied when you use the **File Open**, **File View**, **File Save As** and **File Name** commands, click the **File Type** button, and either select the file type from the adjoining list, or type a new one into the edit control. When you enter a file type, you must always type the leading "**.**"

   To set default modes for file types not in the list set above, click the **Other Types** button.

2. Check and uncheck the mode boxes in the rest of the dialog to specify the modes you want. The meaning of the various mode boxes are detailed below.

3. Click the **Apply** button to store the modes with the particular sort of file chosen in step **1**. You can set modes for other sorts of files by returning to step **1**.

4. If you want to record the mode settings permanently, click the **Save** button.

The modes you can set in the dialog are these:

The Display/Input Modes Area:

This area contains settings that control how the text is displayed in the window, and any special actions PFE takes as you type.

Check the **Auto Indenting** box to have PFE indent new lines to the same level as the preceding line

Check the **Strip Trailing Spaces** box to have white space removed from the end of a line whenever you press **Enter**

Check the **Show Line Numbers** box to display line numbers in the window.

Check the **Overwrite** box to have the characters you type overwrite those under the caret rather than being inserted in front of them.

To associate a language type with the window, select the one you require from the **Language** list.

The Screen Formatting Area:

This area contains settings that affect TABs and text wrapping.

The **Tab Size** edit control lets you define the width of a tab stop.

If you want PFE to insert spaces when you press the **TAB** key, check the **Soft Tabs** box. If the box is

unchecked, PFE inserts a single TAB character.

Check the **Wrap Enabled** box to activate text wrapping. With this turned on, PFE will automatically break the text you type at the column specified in the **Wrap Column** edit control.

The Printing Area:

This area contains settings that affect how PFE prints the text in the current window. These settings become the defaults for the **File Print** command, but you can override them at the time of printing.

Check the **Page Headers** box if you want each page to be headed with the file name, the page number and the date.

Check the **Wrap Long Lines** box if you want lines that are too wide to fit the page to be wrapped to the next line on the page.

The File Modes Area:

This sets the file modes that apply to file, in all the windows that are showing it.

Check the **Read Only** box to make the file *read only*. With this mode set, you will not be able to change the file in any window showing it.

Check the **Save In Unix Format** box if you want all the commands that write the file to disk to do so in Unix format, using a single line-feed character as a line terminator.

Check the **Backup When Saving** button to make a backup copy of any existing file of the same name when you write the file to disk

Check the **Strip Ctrl+Z On Load** button if you want PFE automatically to remove any **Ctrl+Z** character that is the last character of the file when it is loaded

Check the **Add Ctrl+Z On Save** button if you want PFE automatically to add a **Ctrl+Z** character to the end of the file on disk when it is saved.

Check the **No EOLN at end** button if you do not want PFE to write an end-of-line terminator (**CR-LF** or **LF**) after the last character of the last line when the file is saved

The Options Current File/Window Modes Dialog

This dialog lets you set the window modes that apply to the current window, and the file modes that apply to the file that's showing in it.

SETTING THE WINDOW MODES

The **Current Window** area sets the window modes that apply to only the current window.

The Display/Input Modes Area:

This area contains settings that control how the text is displayed in the window, and any special actions PFE takes as you type.

Check the **Auto Indenting** box to have PFE indent new lines to the same level as the preceding line

Check the **Strip Trailing Spaces** box to have white space removed from the end of a line whenever you press **Enter**

Check the **Show Line Numbers** box to display line numbers in the window. You can also change this mode from the tool bar

Check the **Overwrite** box to have the characters you type overwrite those under the caret rather than being inserted in front of them. You can also change this setting from the status bar or with the **Ins** key.

To associate a language type with the window, select the one you require from the **Language** list.

The Screen Formatting Area:

This area contains settings that affect TABs and text wrapping.

The **Tab Size** edit control lets you define the width of a tab stop.

If you want PFE to insert spaces when you press the **TAB** key, check the **Soft Tabs** box. If the box is unchecked, PFE inserts a single TAB character.

Check the **Wrap Enabled** box to activate text wrapping. With this turned on, PFE will automatically break the text you type at the column specified in the **Wrap Column** edit control. You can turn also text wrapping on and off using the status bar.

The Printing Area:

This area contains settings that affect how PFE prints the text in the current window. These settings become the defaults for the **File Print** command, but you can override them at the time of printing.

Check the **Page Headers** box if you want each page to be headed with the file name, the page number and the date.

Check the **Wrap Long Lines** box if you want lines that are too wide to fit the page to be wrapped to the next line on the page.

SETTING THE FILE MODES

The **Current File** area sets the file modes that apply to the current file, in all the windows that are showing it.

Check the **Read Only** box to make the file *read only*. With this mode set, you will not be able to change the file in any window showing it. You can also change this mode using the status bar.

Check the **Save In Unix Format** box if you want all the commands that write the file to disk to do so in Unix format, using a single line-feed character as a line terminator. You can also change this mode using the status bar, and from the dialog started by the **File Save As** command. You cannot change the setting of this mode if the file is marked as *read only*.

Check the **Backup When Saving** button to make a backup copy of any existing file of the same name when you write the file to disk

Check the **Add Ctrl+Z On Save** button if you want PFE automatically to add a **Ctrl+Z** character to the end of the file on disk when it is saved.

Check the **No EOLN At End** button if you do not want PFE to write an end-of-line terminator (**CR-LF** or **LF**) after the last character of the last line when the file is saved

The Edit Goto Line Dialog

This dialog lets you move to the start of an arbitrary line in the current window. You can move either to an absolute line number, or move up or down in the window by a given number of lines.

Steps

1. Enter the line number of the line you want to move to in the **Line To Go To** box. The dialog shows you the number of the last line in the current file, and of the line the caret is now in, for information.

   To move to the first line in the file, enter the number "**1**".

   To move to the last line in the file, you can give the actual line number, or type the word "**end**".

   To move a number of lines up or down from the line that the caret is now in, enter the number of lines to move, preceded by either a "**+**" or a "**-**" sign respectively.

2. If you want to highlight all the text from where the caret now is and the start of the target line, check the **Extend Selection** box

3. Click the **OK** button to go to the target line

The Edit Text Insert ASCII Code Dialog

This dialog lets you insert a character specified by its ASCII code number into the current file at the position of the caret

Steps

1. Specify the character you wish to insert.

   To insert a standard control character such as **ESC** or **Ctrl+Z**, click on the **Control Char** button and select the mnemonic of the control character you want from the list to its right. If you wish to set the top bit (bit 7) of the character, check the **Top Bit Set** box

   To insert a character with an arbitrary ASCII code, click on the **ASCII code** button and type the code as a decimal number in the edit control to its right. You can enter any number between 1 and 255.

2. Click the **OK** button to insert the character with the specified ASCII code at the position of the caret

The File Print Dialog

This dialog lets you print some or all of the current file on your currently selected printer

Steps

1. To use a different printer from the one shown at the top of the dialog box; to change the printer font; or to change the page margins, click the **Setup Printer** button

2. Decide what part of the file you want to print by using the **Print** area:

   Click **Whole File** (the default) to print everything

   Click **Selected Text** to print exactly the text that is highlighted. This option will not be available if you don't have any text highlighted

   Click **Line Range**, and fill in the start and end line numbers, to print a range of lines. You can use the words "**start**" and "**end**" to represent the first and last lines of the file

3. If you want to use different print options from those set by the window modes of the current window, click the boxes in the **Options** area:

   If **Number Lines** is checked, lines are printed with numbering

   If **Wrap Long Lines** is checked, lines too long to fit the page are folded rather than being truncated

   If **Page Headers** is checked, each page starts with a header giving the file name and other information

The File Print Setup Dialog

This dialog lets you set up details of the printer that PFE is to use, and to run the printer's own setup dialog. Settings you make in this dialog are recorded and become the defaults.

You can set different values for each of the printers configured on your system.

Steps

1. If you want to use a different printer to the one that is highlighted in the **Available Printers** list, scroll the list and click the left button on the printer name

2. To change the margins used on each page, check or uncheck the boxes in the **Print Options** area. All margins are 0.5 inches wide

3. To run the highlighted printer's own setup dialog, click the **Setup** dialog.

4. To change the printer font that PFE will use for the highlighted printer, click the **Font** button

If you run the printer's own setup dialog *after* selecting a font, you may find that the font is no longer available; some printers offer different fonts in different operating modes.

The Template Create File Dialog

This dialog allows you to create a new, empty template file that you can use to store templates.

Template files have a file type of "**.tpl**"

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that you want to create the template file in
2. Type the name you want to call the template file, or select an existing name from the **Files** list. If you do specify an existing file, PFE will prompt you to confirm that you really want to overwrite it.
3. Click the **OK** box to create the file

If you prefer, you can simply type the name of the file to be attached directly into the **File Name** edit control rather than use the browse facilities.

About Templates

The Template Edit Dialog

This dialog allows you to edit a template contained in a template file. You must first have attached the template file, either automatically or with the **Template Attach File** command.

Steps

1. Select the template file containing the template you want to edit from the **Template Files** list. As you select a file, the list of templates it holds will appear in the **Template Names** list.
2. Select the name of the template to edit from the **Template Names** list
3. Click the **OK** button to edit the template and close the dialog

PFE opens a new edit window and copies the contents of the template into it for you to change as you require.

When you've finished amending the template, you can update the in-memory copy of the template file with the **Template Store** command, then save the changed template file to disk with the **Template Save File** command.

About Templates

The Template Store As Dialog

This dialog allows you to store a template that you're editing into any attached template file, giving it a new name in the process.

Steps

1. Select the template file into which you want to store the template from the **Template Files** list on the left. The list of templates already in this template file will be updated.

2. Type the name you want to give the template into the **Template Name** edit control on the right, or select the name of an existing template. A template name can be up to 16 characters long, and may contain spaces.

3. Click the **OK** button to store the template in the in-memory copy of the template file, under the new name, and close the dialog.

   If the name you're using is that of an existing template, PFE will ask you to confirm that you really want to overwrite it.

Note that this operation changes only the in-memory copy of the template file. The file on disk will not be updated until you use the **Template Save File** command.

About Templates

The Template Delete Dialog

This dialog allows you to delete one or more templates from a template file. You must first have attached the template file, either automatically or with the **Template Attach File** command.

Steps

1. Select the template file containing the templates you want to delete from the **Template Files** list. As you select a file, the list of templates it holds will appear in the **Template Names** list.

2. Select the name of the templates to delete from the **Template Names** list. You can use **Ctrl** to add names to those selected, and **Shift** to select a range of names. You can also click and drag over a range of names.

3. Click the **Delete** button to delete the templates from the in-memory copy of the template file and close the dialog. To update the template file on disk, you need then to use the **Template Save File** command.

About Templates

The Template Detach File Dialog

This dialog allows you to detach one or more attached template files, freeing main memory, when you no longer require them

Steps

1. Select one or more template files from the list of those attached. You can use the **Ctrl** key to add files to the selection; or the **Shift** key to extend a selection. You can also select multiple files by clicking and dragging.

2. Click the **OK** button to detach the template files and close the dialog. If you've changed any of the templates contained in a template file, you'll be asked if you want to save the changes to disk or to discard them.

Once you've detached a template file, you can't use the templates it contains until you attach it again.

You can't detach a template file if you're currently editing any template that it contains.

About Templates

The Template Save File Dialog

This dialog allows you to save changes that you've made to one or more attached template files to disk, so that they become permanent.

Steps

1. Select the template files you want to save from the list. You can use **Ctrl** to add filenames to the selection, and **Shift** to select a range. You can also click and drag over a range of names.

2. Click the **Save** button to save the in-memory copies of the selected template files to disk and close the dialog. The previous contents of the template files will be overwritten

About Templates

The Template Insert Dialog

This dialog allows you to insert a template from any attached template file into the current window.

Steps

1. Select the template file containing the template you want to insert from the **Template Files** list. As you select a file, the list of templates it holds will appear in the **Template Names** list.
2. Select the name of the template to insert from the **Template Names** list
3. Click the **OK** button to insert the template and close the dialog

The template will be inserted into the current window at the position of the caret. The caret will be positioned at the start of the inserted template data after this operation

About Templates

The Key Mapping Save As Dialog

This dialog allows you to save changes that you've made to the current key mappings into a file of your choice

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that you want to save the key mapping file into

2. Either type the name of the file you want to save the data into in the **File Name** edit control, or select the name of an existing file you want to overwrite from the list box. Key mapping files should normally be given a file type of "**.key**"

3. Click the **OK** box to save the current key mappings to the file. If you've specified the name of a file that already exists, PFE will ask you to confirm that you really intend to overwrite it.

The Key Mapping Load Dialog

This dialog allows you load a set of key mappings from a file of your choice

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that the key mapping file is in

2. Select the file you want to open in the **File Name** list box. Normally, key mapping files have a file type of "**.key**"

3. Click the **OK** box to load the key mappings contained in the file

If you prefer, you can simply type the name of the file containing the key mappings directly into the **File Name** edit control rather than use the browse facilities.

The File Open Dialog

This dialog allows you to open one or more files that already exist.

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that the files are in

2. Select the files you want to open in the **File Name** list box. You can use the **Ctrl** key to add filenames to those selected, and the **Shift** key to select a range of filenames. You can also click and drag across a sequential series of names.

3. If you want to open the files in *read only* mode, check the **Read Only** box. This will affect *all* the files you open in this operation.

4. Click the **OK** box to open the files

If you prefer, you can simply type the name of the file or files to open directly into the **File Name** edit control rather than use the browse facilities.

You can repeat steps **1** and **2** to open files from several drives and directories in the same operation.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

Whether you close the dialog with the **OK** or the **Cancel** buttons, PFE will make its current working directory the one that the dialog is showing.

If you're already editing one of the files you specify, PFE won't load a new copy from disk, but will simply activate a window showing the file.

The File Save As Dialog

This dialog allows you to save the file in the current window to disk, in a file whose name you specify. The name you give for the disk file is then associated permanently with it the current window and all others showing the same file,

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that you want to save the file to

2. Either type the name of the file you want to save the data into in the **File Name** edit control, or select the name of an existing file you want to overwrite from the list box.

3. If you want to save the file in Unix format, check the **Save In Unix Format** box. For normal DOS format, make sure that the box is unchecked.

   The setting you use here becomes the default for the next time you save this specific file.

4. Click the **OK** box to save the data to the file. If you've specified the name of a file that already exists, PFE will ask you to confirm that you really intend to overwrite it.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

After you've saved the current file with this dialog, PFE will associate the name of the disk file with all the windows showing it. The **File Save** command will then write to this file automatically.

The File View Dialog

This dialog allows you to open one or more files that already exist in *read only* mode. Its action is exactly as if you used the **File Open** command and checked the **Read Only** box.

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that the files are in

2. Select the files you want to open in the **File Name** list box. You can use the **Ctrl** key to add filenames to those selected, and the **Shift** key to select a range of filenames. You can also click and drag across a sequential series of names.

3. Click the **OK** box to open the files

If you prefer, you can simply type the name of the file or files to open directly into the **File Name** edit control rather than use the browse facilities.

You can repeat steps **1** and **2** to open files from several drives and directories in the same operation.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

Whether you close the dialog with the **OK** or the **Cancel** buttons, PFE will make its current working directory the one that the dialog is showing.

If you're already editing one of the files you specify, PFE won't load a new copy from disk, but will simply activate a window showing the file.

The File Name Dialog

This dialog allows you to change the file name associated with the file in the current window. The **File Save** command will then write to this file automatically.

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that you want to use in the file name

2. Either type the name of the file you want to save the data into in the **File Name** edit control, or select the name of an existing file from the list box.

3. Click the **OK** box to associate the new file name with the file in the current window

If you prefer, you can simply type the name of the file directly into the **File Name** edit control rather than use the browse facilities.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

The File Write Dialog

This dialog allows you to save the file in the current window to disk, in a file whose name you specify. Unlike the **File Save As** command, the filename associated with the window is *not* changed.

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that you want to write the file to

2. Either type the name of the file you want to save the data into in the **File Name** edit control, or select the name of an existing file you want to overwrite from the list box.

3. Click the **OK** box to save the data to the file. If you've specified the name of a file that already exists, PFE will ask you to confirm that you really intend to overwrite it.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

The File Insert Dialog

This dialog allows you to insert the entire contents of an existing file into the current window at the position of the caret.

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that the file is in
2. Select the file you want to open in the **File Name** list box
3. Click the **OK** box to insert the entire contents of the file

If you prefer, you can simply type the name of the file to be inserted directly into the **File Name** edit control rather than use the browse facilities.

The **List Files of Type** list at lower left allows you to restrict the files shown in the list to those matching specific filename patterns. By default, there is one entry - "***.***" - in this list, so you'll see all the files in each directory. You can configure the list to match how you want to work by editing the initialisation file.

The Edit Find Dialog

This dialog allows you to find occurrences of text strings. Unlike most dialogs, you can continue to work in PFE while it is visible, so you can switch between your windows to search in several of them in one operation.

Steps

1. Type the text you want to search for in the **Find What** edit control. A special notation allows you to search for characters like "end of line" that you can't type. The notation uses the "\" character, so you must type this as "\\".

   If the current window contained a highlighted selection when you started the dialog, the **Find What** edit control will show the selected text, allowing you quickly to search for further occurrences. PFE also records the last 8 strings you searched for, and you can repeat the search for one of these by clicking on the drop-down button to the right of the edit control and selecting the string from the list. The string you searched for last time will always be at the head of the list.

2. In the **Direction** area, set the direction of search. You can search *up* from the caret's current position to the start of the file, or *down* from the caret's position to the end of the file

3. Set up the search options you want to apply:

   If the **Match Case** box is checked, PFE will match what you type exactly. If it's unchecked, upper-case and lower-case letters are taken as the same

   If the **Select Matching Text** box is checked, PFE will highlight any matching text it finds. If it's unchecked, the text won't be highlighted

   If the **Extend Selection** box is checked, PFE will highlight all the text from where the caret is now to the end of the matching text. You can only check this box if you've also checked the **Select Matching Text** box.

4. Click the **Find Next** button to perform the search. You can repeat this as often as you want.

   If you want to search in some other window, simply activate it while this dialog is still on screen and click the **Find Next** button again.

5. To close the dialog, click the **Cancel** button.

After you've performed a search, you can repeat it very quickly with the **Edit Repeat Last Find** command, which will make the search without showing this dialog.

By default the strings saved in the **Find What** list are not recorded for subsequent edit sessions. However, you can request that PFE does record them by setting the **save-find-strings** value in the **[options]** section of the initialisation file.

The Edit Replace Dialog

This dialog allows you to find occurrences of a text string and replace them with another. Unlike most dialogs, you can continue to work in PFE while it is visible, so you can switch between your windows to make changes in several of them in one operation.

The search is always made from the current position of the caret down to the end of the file.

Steps

1. Type the text you want to search for in the **Find What** edit control. A special notation allows you to search for characters like "end of line" that you can't type. The notation uses the "\" character, so you must type this as "\\".

   If the current window contained a highlighted selection when you started the dialog, the **Find What** edit control will show the selected text, allowing you quickly to search for further occurrences. PFE also records the last 8 strings you searched for, and you can repeat the search for one of these by clicking on the drop-down button to the right of the edit control and selecting the string from the list. The string you searched for last time will always be at the head of the list.

2. Type the text you want to replace it with in the **Replace With** edit control. The same notation is used for characters you can't type.

   PFE also records the last 8 strings you used as replacement strings, and you can re-use one of these by clicking on the drop-down button to the right of the edit control and selecting the string from the list. The string you used last time will always be at the head of the list.

   To simply delete occurrences of the search string, leave this edit control blank.

3. Set up the search options you want to apply:

   If the **Match Case** box is checked, PFE will match what you type exactly. If it's unchecked, upper-case and lower-case letters are taken as the same

4. Select an operation you want to perform:

   Clicking the **Find Next** button moves to the next occurrence of the search string

   Clicking the **Replace** button replaces the current match (or the first match it finds after the caret), and moves to the next

   Clicking the **Replace All** button replaces all matches automatically

5. To close the dialog, click the **Cancel** button.

You can repeat step **4** as often as you wish. To work in another window, simply activate it and choose an action from step **4**.

Clicking the **Undo Last** button will reverse the effect of the last replacement made in the current window. To undo the effects of more than one replacement, use the **Edit Undo** menu command or the undo toolbar button.

After you've performed a replace operation, you can repeat it very quickly with the **Edit Repeat Last Replace** command, which will make the replacement without showing this dialog.

By default the strings saved in the **Find What** and **Replace With** lists are not recorded for subsequent

edit sessions. However, you can request that PFE does record them by setting the **save-find-strings** value in the **[options]** section of the initialisation file.

The Template Attach File Dialog

This dialog allows you to *attach* a template file, making the templates that it contains available for use.

Template files have a file type of "**.tpl**"

Steps

1. Use the **Drive** and **Directory** list boxes to select the disk drive and the directory that the template file is in
2. Select the file you want to open in the **File Name** list box
3. Click the **OK** box to attach the file

If you prefer, you can simply type the name of the file to be attached directly into the **File Name** edit control rather than use the browse facilities.

About Templates

The Execute DOS Command To Window Dialog

This dialog allows you to run a DOS program, such as a compiler, or an internal command such as **dir**, and capture the output into an edit window.

You can run only one DOS program at a time with this dialog.

You can't start a Windows application with this dialog; for that, use the **Execute Launch Application** command.

Steps

1. Type the DOS command line you want to execute into the **Command** edit control. You can automatically substitute parts of the name of the file showing in the current window in the command line if you wish. Such substitution points are indicated by a "**%**" character, so if you wish to include a "**%**" in the command line unaltered, type it as "**%%**"

   You can also select one of the previous command lines used from the drop down list box.

   To browse your directories to find the program you want to run, click the **Browse** button.

2. Type the path of the working directory you want the command to start in into the **Working Directory** edit control. If you leave this blank, the command starts in PFE's current working directory, which is shown at the top of the dialog.

3. Set the various options you want to apply:

   If the **Beep When Done** box is checked, PFE sounds the standard system beep when the DOS program finishes.

   If the **Minimize Editor** box is checked, PFE will make itself into an icon before running the DOS program

   If the **Save Changed Files** box is checked, PFE will see if any of the files you're editing have changed, and will give you the opportunity to save the changes before running the DOS program. If you agree to save the changes, *all* the files you've altered will be written to disk.

   If the **Reuse Output Window** box is checked, PFE will place the DOS program's output into the window used the last time you used this dialog. If not, a new output window will be created.

   If the **Show End Of Output** box is checked, PFE will automatically scroll the window showing the output to show you the end of it rather than the start.

4. Click the **OK** button to run the DOS program.

The command line, working directory path and options are recorded, and become the default for the next time you use this dialog.

Under Windows 3, to change the properties of the DOS window used to run the program, use the standard **PIF Editor** to modify the PIF file **$pfedos.pif**

To repeat the operation exactly, use the **Execute Repeat DOS Command to Window** command.

The Execute Launch Application Dialog

This dialog allows you to launch a Windows application, or a DOS program, to run independently of PFE.

You can't capture the output of a DOS program launched with this dialog. For that, use the **Execute DOS Command To Window** command.

Steps

1. Type the command line for the application you want to launch you want to execute into the **Command** edit control, or select one of the previous command lines you used from the drop-down list box

   To browse your directories to find the application, click the **Browse** button.

2. Type the path of the working directory you want the application to start in into the **Working Directory** edit control. If you leave this blank, the application starts in PFE's current working directory, which is shown at the top of the dialog.

3. Set the various options you want to apply:

   If the **Minimize Editor** box is checked, PFE will make itself into an icon before running the DOS program

   If the **Save Changed Files** box is checked, PFE will see if any of the files you're editing have changed, and will give you the opportunity to save the changes before launching the application. If you agree to save the changes, *all* the files you've altered will be written to disk.

4. Click the **OK** button to launch the application.

The command line, working directory path and options are recorded, and become the default for the next time you use this dialog.

The Execute Launch Windows Tool Dialog

This dialog lets you launch one of the Windows Tools that you've configures in the list of Windows Tools with the **Execute Configure Windows Tools** command.

Steps

1. Select the name of the tool that you want to launch from the list at the lower left of the dialog. The command line and options that you defined for the tool will be shown.

2. If you wish, modify the details of the command line shown in the **Command Line** edit control. .

3. If you wish, modify the options that apply in the **Options** area:

   If the **Minimize Editor** box is checked, PFE will make itself into an icon when you launch the tool

   If the **Save Changed Files** box is checked, PFE will see if you have changed any of the files you're editing before launching the tool, and give you a chance to save them.

4. Enter the path name of the directory that you wish the tool to run in into the **Working Directory** edit control. If you leave this field blank, the tool will be run in PFE's current working directory, which is shown at the top of the dialog.

5. Click the **OK** button to launch the tool.

The values you set in this dialog are recorded, and will be used as the defaults when next you launch the same tool. The **Reset Command Line** button can be used at any time to restore the command line to the one you defined originally.

The Execute Configure Windows Tools Dialog

This dialog lets you add entries to the list of Windows tools, and to delete or change existing entries.

About Windows Tools

ADDING A NEW ENTRY

To add a new entry to the list of Windows tools, do this:

Steps

1. Type the name you want to use for the tool in the **Tool** list at the top of the dialog. A tool name can be up to 32 characters long, and can contain spaces.

2. Type the command line you want to associate with the tool name in the **Command Line** edit control. You can browse your disks and directories to find the name of the executable module by clicking the **Browse** button.

3. Set the default options you want to apply to the tool name in the **Options** area:

   If the **Minimize Editor** box is checked, PFE will make itself into an icon when you launch the tool

   If the **Save Changed Files** box is checked, PFE will see if you have changed any of the files you're editing before launching the tool, and give you a chance to save them.

4. Click the **Add** button to add the tool name and details to the list.

5. If you have more tools to add, return to step **1**; or follow the procedures below to change or delete other entries. When you've finished, click the **Save** button if you want to record the amended list permanently.

DELETING AN EXISTING ENTRY

To delete an existing entry from the Windows Tools lists, do this:

Steps

1. Select the name of the tool you want to delete in the **Tool** list at the top of the dialog.

2. Click the **Delete** button to delete the entry

3. If you have more tools to delete, return to step **1**; or follow the other procedures given here to add or change other entries. When you've finished, click the **Save** button if you want to record the amended list permanently.

CHANGING AN EXISTING ENTRY

To alter some or all of the details of an existing entry in the Windows Tools list, do this:

1. Select the name of the tool whose details you want to amend in the **Tool** list at the top of the

dialog.

2 If you want, modify the command line you want to associate with the tool name in the **Command Line** edit control. You can browse your disks and directories to find the name of the executable module by clicking the **Browse** button.

3 If you want, modify the default options you want to apply to the tool name in the **Options** area:

   If the **Minimize Editor** box is checked, PFE will make itself into an icon when you launch the tool

   If the **Save Changed Files** box is checked, PFE will see if you have changed any of the files you're editing before launching the tool, and give you a chance to save them.

4 Click the **Change** button to change the details associated with the tool name to those now set in the dialog.

5 If you have more tools to change, return to step **1**; or follow the procedures above to add or delete other entries. When you've finished, click the **Save** button if you want to record the amended list permanently.

The Options Key Mapping Dialog

This dialog allows you to alter the functions that are invoked when you press certain keys. The dialog allows you map new keys to functions; to quickly change a key mapping; and to configure prefix keys.

To map a key to a function, you can begin either with the key code, or with the name of the function. If you want to map a specific key to a function, it's easiest to begin by specifying the key; if you want to change the keys that map to a given function, it's best to start with the function name. However, you can use either method as you prefer.

TO MAP A KEY TO A FUNCTION

If you want to map a specific key to a function, it's best to start by specifying the key. Take the steps given here:

1. If the key you want to map is a prefix key, select the prefix key from the drop-down list at the top of the dialog. If the prefix key you want to use does not appear in the list, click the **Configure** button to run the prefix key configuration dialog. You can use **Esc** or any control key from **Ctrl+A** to **Ctrl+Z** as a prefix key.

2. Specify the second (or only) key to press from the controls in the next row of the dialog

   Check the **Alt** box to specify that the **Alt** key must be held down

   Check the **Ctrl** box to specify that the **Control** key must be held down

   Check the **Shift** box to specify that the **Shift** key must be held down

   Then select the key itself from the drop-down list on the right.

   If the key can be mapped to a function, the name of the key will appear in the box below as confirmation. PFE will also indicate whether or not the key is already mapped to a function.

3. In the **Functions** section at the bottom of the dialog, select the name of the function you want to map the key to from the drop-down list. PFE will show you the names of any keys already mapping to this function in the list below the name

4. If you want the key name to appear on the menu associated with this function, check the **Show this mapping on menu** box; uncheck it if you do not want the key name to appear. This option won't be available if the function you've chosen has no corresponding menu item

5. Click the **Add/Change** button at the top right of the dialog to map the selected key to the selected function

TO UNMAP A KEY FROM A FUNCTION

To unmap a key from a function, you can if you like use the steps detailed above and start by specifying the key. However, it's probably easier if you use the steps below and start with the function name.

1. In the **Functions** section of the dialog, select the name of the function from the drop-down list. PFE will show the names of the keys currently mapping to this function in the list below the function name

2 Select the key whose mapping you want to delete by double-clicking the left mouse button on the name. The key details will appear at the top of the dialog.

3 If you simply want to unmap the key so it does nothing, click the **Delete** button.

If you want to map the key to a *different* function, select the new function name from the drop-down list in the **Commands** section of the dialog, and click the **Add/Change** button.

SAVING CHANGES

Changes you make in this dialog apply only to the current PFE session unless you save them.

To save the changes so that they become the default key mapping that PFE will use when it starts, click the **Save** button. To avoid the risk of accidental change, if you specified a key map file when you started PFE with a "**/k**" command line option, the **Save** button will not be available. In this case, you must always use the **Save As** button to save changes.

To save the changes into a key map file, so that you can manually load and apply them when you wish, click the **Save As** button. This starts a sub-dialog that lets you specify the name of a file into which the key mappings can be saved.

LOADING A SET OF KEY MAPPINGS

To load a set of key mappings that you've saved to a file with the **Save As** button, click the **Load** button. This starts a sub-dialog that lets you specify the name of a file containing mappings, which PFE will load and apply.

The Options Prefix Keys Dialog

This dialog lets you define which keys are used as *prefix keys*, or keys which start two-character command sequences. You can use the **Esc** key, or any control key from **Ctrl+A** to **Ctrl+Z** as a prefix key

About Key Mapping

ACTIVATING KEYS AS PREFIXES

To make keys that are currently not prefixes act as prefixes, select the key names from the **Inactive prefix keys** list on the left of the dialog, and click the **Activate** button. The names will transfer to the **Active prefix keys** list on the right.

Note that if you make a key an active prefix, any mappings you've set up for it *by itself* will be suspended.

DEACTIVATING KEYS AS PREFIXES

To make keys that are currently prefixes act as ordinary keys, select the key names from the **Active prefix keys** list on the right of the dialog, and click the **Deactivate** button. The names will transfer to the **Inactive prefix keys** list on the left.

The Options Screen Font Other Dialog

This dialog allows you to choose the screen font that PFE will use to display text in all its edit windows. You can choose any fixed-pitch font that's installed in your system.

Steps

1. Select the name of the font you want to use from the **Font** list on the left.

2. If you want some other style than plain text, such as italic or bold, select this from the **Font Style** list in the centre of the dialog.

3. Select the size of the font, in points, from the **Size** list at the right. If the font you've selected is a *scalable* font - such as a TrueType font or one generated by Adobe Type Manager - you can choose any size between 6 and 24 points. Other fonts may offer only certain sizes.

4. Click the **OK** button to select the font and close the dialog.

PFE will use the font details you've chosen for all windows. The details are recorded and become the defaults.

The Printer Font Dialog

This dialog allows you to choose the printer font that PFE will use to print text on the currently-selected printer, whose name is shown at the top of the dialog. You can choose any fixed pitch font that the printer supports.

Steps

1. Select the name of the font you want to use from the **Font** list on the left.

2. If you want some other style than plain text, such as italic or bold, select this from the **Font Style** list in the centre of the dialog.

3. Select the size of the font, in points, from the **Size** list at the right. If the font you've selected is a *scalable* font - such as a TrueType font or one generated by Adobe Type Manager - you can choose any size between 6 and 24 points. Other fonts may offer only certain sizes.

4. Select the type face effects you want from the **Effects** area:

   Check the **Strikeout** box to have the text printed with a strikeout line through it

   Check the **Underline** box for underlined text

5. If your printer supports colour, select the type colour from the **Color** box.

6. Click the **OK** button to select the font and close the dialog.

PFE will use the font details you've chosen whenever you print to that printer. The details are recorded and become the defaults for the printer.

The details of the printer font set up for any other printers will not be affected.

The Exit Windows Dialog

This dialog allows you to close PFE and also terminate Windows 3 at the same time.

Steps

1. Select the way you want to close Windows

   Click the **Exit To DOS** button to close Windows and return to a DOS prompt

   Click the **Exit And Restart Windows** button to close Windows and then automatically restart it with the same options that you used previously

   Click the **Exit And Reboot System** button to close Windows and perform a warm reboot of your system

   Click the **Exit, Run DOS Command And Restart** button to close Windows, execute a single DOS program or batch file, then restart Windows. If you choose this button, enter the command line to run in the edit control below it. Note that you must give a full pathname to the **.EXE** or **.BAT** file. The **Browse** button will let you browse your disks and directories for the program to run.

   If you're running PFE for Windows/16 in the Win16 subsystem of Windows NT, all four options will simply log you off.

2. Click the **OK** button. If any of the files you're editing have been changed, PFE will ask if you want to save the changes; you can save them, discard them, or abandon the closedown altogether.

Note that if you select the option to run a DOS command and restart Windows, and the executable is not found, or cannot be run for any reason, you will see no notification of the error.

The Exit Windows NT Dialog

This dialog allows you to close PFE and either log off or close Windows NT

Steps

1. Select the Windows NT closedown action you want to perform

   Click the **Logoff User Only** button to simply end all the applications you've started and log you off

   Click the **Logoff User And Shutdown System** button to act as above, then close Windows NT down so that you can power off your system

   Click the **Logoff User And Reboot System** button to log you off, close down Windows NT and perform a warm reboot of your system

   You need to have **SE_SHUTDOWN** privilege for all but the first option.

2. If you want to force running applications that do not respond to close automatically, rather than having Windows NT prompt you for each one, check the **Force Other Apps To Close** box.

3. Click the **OK** button. If any of the files you're editing have been changed, PFE will ask if you want to save the changes; you can save them, discard them, or abandon the closedown altogether.

The Browse Dialog

This dialog allows you to browse your disks and directories to find a file name that can be used by its parent dialog.

Steps

1. Use the **Drives** and **Directories** lists to move around your files until you find the directory containing the file you're interested in.

2. Select the name of the file from the **File Name** list on the left.

3. Click the **OK** box to close the dialog. The full path name of the file you've selected will appear in the appropriate edit control of the parent dialog.

The Select Window Dialog

This dialog allows you to select between many open edit windows. The list at the top of the dialog shows the window captions of some or all of the windows.

Steps

1. From the **Windows Showing** area, select the sort of window you want to see in the list:

   Check the **Named Files** box to see windows showing files that have associated file names

   Check the **Unnamed Files** box to see windows showing files that do not have associated file names

   Check the **Templates** box to see windows showing templates

   Check the **Command Output** box to see windows showing the output from DOS commands

2. If you want to see only windows containing changes that have not yet been saved, check the **Changed Windows Only** box

3. Either double-click the left mouse button on the name of the window you want to activate, or select it and press the **OK** button

The settings of the check boxes are remembered, and become the default for the next time you use the dialog.

The Most Recently Used Files Dialog

This dialog allows you to select a file from the list of those you've used most recently, and open it.

Steps

1. If you want to change the order in which the file names are presented in the list at the top of the dialog, click the appropriate button in the **Sort Order** area:

   Click the **Last-used First** button to order the list so that the file you opened most recently is at the top

   Click the **Alphabetical** button to sort the list in alphabetical order

2. If you want to see the full path name of every file in the list, check the **Show Full Pathnames** box. If this box is un-checked, PFE will shorten filenames as far as possible, making them relative to the current directory.

3. To open a file from the list, either double-click its name in the list with the left mouse button, or select the name and press the **Open** button.

The settings of the check boxes and sort buttons are remembered, and become the default for the next time you use the dialog.

Save Changes To File?

You have changed this file since the last time you saved it to disk. Before PFE closes it, you have a chance to update the disk copy.

Yes Click this button to save the contents of the window to disk. If the file hasn't yet got a name, you'll see a dialog asking you to specify the file to save to.

No Click this button to irretrievably discard the changes you've made since the last save.

Cancel Click this button to abandon the operation completely and return to editing the window.

Save Changes To Template?

You have changed this template since the last time you stored it in the in-memory copy of a template file. Before PFE closes it, you have a chance to update the in-memory template file.

Yes Click this button to save the contents of the window to an in-memory copy of a template file. If the template hasn't yet got a name, you'll see a dialog asking you to specify what to call it, and what template file you want to store it in.

No Click this button to irretrievably discard the changes you've made since the last save.

Cancel Click this button to abandon the operation completely and return to editing the window.

Storing a template in an in-memory copy of a template file does not update the template file on disk; you need to use the **Template Save File** command for this.

Cannot Detach Template File

You cannot detach a template file if you are editing one or more of the templates it contains.

Steps

1. For each of the windows containing such templates, use the **Template Store** or **Template Store As** commands to save changes you want to keep, then close the window with the **Window Close** command.

2. Preserve the changes to the template file itself with the **Template Save File** command

You will then be able to detach the template file.

Invalid Line Number

You have entered an invalid line number in the **Line To Go To** edit control. Check that what you have typed is either a numeric string, or is the word **end**.

If you have typed a number preceded by either **+** or **-** to specify a line number relative to the line that the caret is now in, check that the resulting line number is not less than 1, or greater than the number of the last line in the file

Invalid ASCII Code

You have entered an invalid ASCII code number in the dialog. Make sure that what you type is a decimal number, in the range 1 to 255.

Default Printer Not Configured

Whenever you exit from PFE, the name of the printer you last used is recorded in its initialisation file. This device is used as PFE's default printer the next time you start the program.

While starting this session, PFE has detected that the printer/port combination you used last time is no longer configured in your system. PFE will now use the Windows default printer unless you specify otherwise. You can check that this is what you want, and correct it if not, by using the **File Print Setup** command.

Replaying A Keyboard Macro While Recording

You have attempted to replay the keyboard macro that you are currently recording, which is not permitted.

You can choose to switch off the keyboard macro recorder now by clicking the **No** button. If you click the **Yes** button PFE will continue to record your keystrokes. This failed action will not be stored in the keyboard macro, and will not be repeated when you replay it.

Create A Further Backup Level

PFE has detected that the file you are about to overwrite in this operation is already in a directory that contains backups of overwritten files. You may not wish to take a further backup of the file before it is overwritten.

If you click the **Yes** button, PFE will continue as normal and take a backup copy of the file before saving the new data to it. This will involve creating a new sub-directory (called by default "**$PFEBK**") to contain it; depending on how deep your directory structure is at this point, the full path name of the directory, or of the backed up file within it, may exceed the limits imposed by the operating system.

If you click the **No** button, PFE will *not* take a backup copy of the existing file, but will simply overwrite it with the new data. The current contents of the disk file will be irrevocably lost.

If you click the **Cancel** button, the save operation will be abandoned and the disk file will not be changed.

Taking Backups Of Saved Files

Cannot Perform Command Line Substitution

The command line that you have asked PFE to execute contains substitution characters, which normally would be replaced by parts of the name of the file showing in the current window.

PFE cannot perform the requested substitutions, because either you have no files open, or because the current window does not contain a file with an associated name. You cannot substitute parts of the names of templates; and command output windows and those created by the **File New** command do not have filenames associated with them.

Command Line Is Too Long

While substituting parts of the name of the file showing in the current window into the command line to be executed, the resulting command became too long for the 512-byte buffer that PFE uses to store it.

You should change the command so that it does not exceed this size. Note that although PFE allows up to 512 bytes for the string, this may exceed the size allowed by the operating system or command processor that will run the command.

Invalid Substitution Point Encountered

While substituting parts of the name of the file showing in the current window into the command line to be executed, PFE encountered an unrecognised substitution symbol and could not continue.

Check the command line you have specified and correct the invalid substitution request. All substitution points refer to parts of the full path name of the file showing in the current window, and are these:

%d	The directory part of the path name, without a trailing "****"
%e	The extension part of the path name, without a leading "**.**"
%f	The filename and extension parts of the path name, separated by a "**.**"
%n	The filename part of the path name
%p	The entire path name
%u	The drive part of the path name, followed by a "**:**"
%%	A single "**%**" character

Clear Undo Actions

This command will clear the details of all the previous editing actions that you have recorded for the current file.

If you click the **NO** button, PFE will abandon the action and no change will be made

If you click the **YES** button, PFE will clear the details it has recorded. This will result in some memory being freed for other uses (depending on what editing actions you have performed); but you will not be able to undo any changes you have made up to this point.

Context Help Not Available

PFE is unable to give context help on the item selected in the current window, because you have not specified the help file that is to be searched.

You can specify the help file in two ways. PFE will use for preference the help file that you have defined with the **help-context-file** key in the **[options]** section of the initialisation file.

If you have not specified a file in this way, PFE will use the first *user defined* help file named in the **[help-files]** section of the initialisation file.

The Status Bar

This area of the screen is the **status bar**, which provides you with useful information on the state of PFE itself and on the current file.

For a full description of the status bar, select the **About the status bar** topic at the end of this item.

To see interactive help on any of the boxes within the status bar, use the **Help Screen/Menu Help** command to turn on interactive help, and click the left mouse button once with the box of interest.

About the Status Bar

Status Bar - File Position Area

This area of the status bar reports the position of the caret in the current file, in terms of the line number within the file, and the column within the line.

Double-clicking the left mouse button in this area will execute an **Edit Goto Line** command, which starts a dialog that lets you move quickly to an arbitrary line in the file.

If you don't have any files open, this area shows you the PFE's version number.

Status Bar - Lines In File Area

This area of the status bar tells you how many lines there are within the current file. If you don't have any files open, it will be blank.

When a file is being loaded, the number in this area will steadily increase, to show you how many lines have been processed so far. When you save a file, the number will decrease to zero, showing you how many lines are still to be written to disk

Status Bar - File Change Marker

This area of the status bar tells you whether the current file has been changed. If it has, the area will display a '**#**' character; if not, or if you don't have any files open, the area will be blank.

Status Bar - File Read-Only Marker

This area of the status bar tells you whether the current file has been made *read-only*, and so cannot be altered.

If the file is read-only, the area will show the characters "**RO**". If you are able to change the file, it will show "**RW**".

Double-clicking the left mouse button in this area will change the file's state from *read-only* to *writable* and vice-versa.

Status Bar - Prefix Key Area

This area of the status bar tells you whether you have started a two-key command sequence by typing one of the configured prefix keys.

If you have, the area shows you the mnemonic of the prefix key you've used. If you haven't started a two-key command, or you have no files open, the area will be blank.

Status Bar - Macro Recording Area

This area of the status bar tells you whether keystrokes are currently being recorded in a keyboard macro.

If they are, the area will contain the text "**Rec on**". If you're not recording keystrokes, it will contain the text "**Rec off**". The area will be blank if you don't have any files open.

Double clicking the left mouse button in this area will turn recording on or off.

Status Bar - Text Wrap Area

This area of the status bar tells you whether text wrapping is active in the current window.

If wrapping is not turned on, the area will show the text "**No wrap**". Otherwise, it will show the word "**Wrap**", followed by the number of the window column at which wrapping will occur.

The area will be blank if you don't have any files open.

Double clicking the left mouse button in this area will turn wrapping on or off.

Status Bar - Save Format Area

This area of the status bar tells you the format in which the current file will be saved if you write it to disk.

If the area shows the text "**DOS**", the file will be written with each line terminated by a **Carriage Return** byte and a **Line Feed** byte (**CR-LF**), which is the standard for MS-DOS files.

If the area shows the text "**Unix**", the file will be written in Unix format, with each line ended by a single **Line Feed** byte (**LF**).

The area will be blank if you don't have a file open.

Double clicking the left mouse button in this area will change the format.

Status Bar - Typing Mode Area

This area of the status bar tells you the typing mode of the current window.

If the area shows the text "**INS**", characters you type will be inserted, displacing any characters already in the file to the right. If it shows the text "**OVR**", characters you type will overwrite those already there.

Double clicking the left mouse button in this area will change between the two modes.

Status Bar - Num Lock Area

This area of the status bar tells you the current state of the **Num Lock** key. It will show the text "**NUM**" if Num lock is active.

Status Bar - Caps Lock Area

This area of the status bar tells you the current state of the **Caps Lock** key. It will show the text "**CAP**" if caps lock is active.

Tool Bar Button

 This tool bar button lets you quickly perform actions such as creating a new file edit window. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Ctrl** or **Shift** keys down at the same time.

Left click alone

This executes a **File New** command to create a new, empty edit window

Left click + Shift

This executes a **Template New** command to create a new, empty template window

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly perform actions such as opening existing files. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Ctrl** or **Shift** keys down at the same time.

Left click alone

This executes a **File Open** command to open an existing file for editing

Left click + Shift

This executes a **File View** command to open an existing file in *read-only* mode

Left click + Ctrl

This executes a **Template Edit** command to edit an existing template

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly perform actions such as writing files to disk. To use it, position the mouse cursor on the button and click the left button once.
This executes a **File Save** command to write the current file to disk. If the current file is a template rather than a file, a **Template Store** command is executed to save it to a template file.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly cut the selected text in the current window to the clipboard. To use it, position the mouse cursor on the button and click the left button once.
This executes an **Edit Cut** command to delete the highlighted text from the current file and transfer it to the clipboard.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly copy the selected text in the current window to the clipboard. To use it, position the mouse cursor on the button and click the left button once.
This executes an **Edit Copy** command to copy the highlighted text to the clipboard. The current file is not changed by this operation.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly paste the contents of the clipboard into the current window. To use it, position the mouse cursor on the button and click the left button once.
This executes an **Edit Paste** command to paste the clipboard into the current window at the position of the caret.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly perform search actions in the current file. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** key down at the same time.

Left click alone

This executes an **Edit Find** command to show you the dialog that controls searching for text

Left click + Shift

This repeats exactly the last action you performed with the **Edit Find** command

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly perform text replacement actions. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** key down at the same time.

Left click alone

This executes an that starts the dialog controlling text replacement actions

Left click + Shift

This repeats exactly the last text replacement operation you performed with the **Edit Replace** command

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly undo the last edit action. To use it, position the mouse cursor on the button and click the left button once.
This executes an **Edit Undo** command to reverse the effects of the last edit action you performed.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly insert a template into the current window. To use it, position the mouse cursor on the button and click the left button once.
This executes a **Template Insert** command, which will show you a dialog from which you can select the template to insert.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly perform actions related to keyboard macros. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** key down at the same time.

Left click alone

This executes a **Macro Start Recorder** command to start recording if not already started; or a **Macro Stop Recorder** command to stop it.

Left click + Shift

This executes a **Macro Replay** command to replay the last-recorded keyboard macro.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly start a DOS command processor session. To use it, position the mouse cursor on the button and click the left button once.
This executes an **Execute DOS Prompt** command to start a command shell

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly initiate DOS commands such as compilers, capturing their output in a window. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** key down at the same time.

Left click alone

This executes an **Execute DOS Command To Window** command, which shows you a dialog allowing you to specify the command line to be executed.

Left click + Shift

This executes an **Execute Repeat DOS Command To Window** command to repeat the last command executed with **Execute DOS Command To Window**.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly launch Windows applications, or configure the list of Windows tools. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** or **Ctrl** keys down at the same time.

Left click alone

This executes an **Execute Launch Application** command, which starts a dialog allowing to specify the command line for the application to be run.

Left click + Shift

This executes an **Execute Launch Windows Tool** command to start a dialog allowing you to launch a pre-configured Windows tool.

Left click + Ctrl

This executes an **Execute Configure Windows Tools** command to start a dialog which allows you to set up the details of the Windows tools you want to use.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly turn line numbering on or off in the current window. To use it, position the mouse cursor on the button and click the left button once.
Line numbering will be turned off it is currently on, and on if it is currently off.

About the Tool Bar

Tool Bar Button

 This tool bar button lets you quickly initiate printer-related actions. To use it, position the mouse cursor on the button and click the left button once, possibly holding the **Shift** key down at the same time.

Left click alone

This executes a **File Print** command which starts a dialog allowing you to print the current file.

Left click + Shift

This executes a **File Print Setup** command which starts a dialog allowing you to configure details of the printer to use.

About the Tool Bar

Recently Used File

This menu item contains the name of a file that you have opened recently. PFE maintains a list of the files you open, and automatically adds them to the end of the **File Menu**.

Clicking the left mouse button on one of these file name items will open the file for editing.

More Files/Recent Files

This menu item is part of the list that PFE maintains of files that you have opened recently. You can configure the number of files that PFE records, and if you set this number to more than PFE can display on the end of the **File Menu**, this menu item will be added.

Clicking the left mouse button on the command will start a dialog that will allow you to choose any file from the complete list of those opened recently.

Active Window

This menu item contains the title of one of the windows that are currently open. PFE automatically adds the titles to the end of the **Window Menu**.

Clicking the left mouse button on one of these window titles will make the corresponding window active.

If you have more than 9 windows open, an additional **More Windows** item will appear on the end of the menu; selecting this item shows you a dialog that lets you select from all the titles.

More Windows

This menu item is added to the end of the **Window Menu** whenever you have more than 9 windows open simultaneously.

Selecting this item shows you a dialog that lets you select from the titles of all the open windows and make one of them the active window.

Exit Windows

This system menu item allows you to terminate PFE, and to shut down your Windows system in a variety of ways.

The command starts a dialog that allows you to specify various closedown options

Save Screen

This system menu item allows you to start a Windows screen saver immediately, instead of having to wait for the configured inactivity period.

This option will not work with screen savers that do not use the standard Windows screen-saver interfaces to function.

The System Menu

The system menu contains a number of standard commands that allow you to manipulate PFE's main window and change to other applications. The items at the start of the list are common to all Windows applications.

Close

Terminates your PFE session

Move

Allows you to move PFE's main window with the keyboard cursor keys

Size

Allows you to resize PFE's main window with the keyboard cursor keys

Minimize

Makes PFE's main window into an icon

Maximize

Expands PFE's main window so it fills the entire screen

Restore

If PFE's main window is an icon, restores it to its previous windowed state

Switch To

Starts the Windows Task Manager to allow you to shift to another application

A word is defined as a sequence of alphanumeric characters, delimited at each end by white space or punctuation. If the window has been set with a window mode of "C language" the underscore character is treated as alphanumeric

To represent characters that you can't type in the **Edit Find** and **Edit Replace** dialogs, and in the search string specified in the **EditFind** DDE command, use this notation:

\f represents a Form Feed character
\t represents a TAB character
\n represents the end of a line
**** represents a single '\' character

To represent an arbitrary character code, use the notation "**\xnn**", where "**nn**" represents two hexadecimal digits. The null value "**\x00**" is not permitted.

PFE can capture the output of any DOS program that writes to the standard output or standard error files **stdout** and **stderr**. Output displayed by writing directly to video memory can't be captured.

MAPI is the electronic mail protocol devised by Microsoft and used in their Microsoft Mail application. Other vendors such as Novell use different mail systems; currently PFE does not support them

To establish a DDE link to the Windows/16 version of PFE, whether under Windows 3.1 or the Win16 subsystem of Windows NT, use the service name **PFE**. To link to the Windows NT version, use the service name **PFE32**

This help file was created for Programmer's File Editor version 0.05.007