

ProgramBar for Windows 3.1

Written by Ian Jefferies

(jeffers@teaching.physics.ox.ac.uk)

Copyright © 1995 Ian Jefferies. All rights reserved.

This release: 1 December, 1995

Version 2.11

Welcome to ProgramBar help!

Inspired by the task bar found in Windows95, this program has been written so that the same kind of functionality and ease of use can be brought to Windows 3.1 and Windows For Work Groups.

This version incorporates many new usability enhancements over version 1.x. The most significant being a built in configuration editor. My thanks to those of you who have written to me expressing your support for ProgramBar, pointing out bugs and suggesting enhancements. All of the bugs have been fixed for this release and many of the enhancements included.

[Introduction to ProgramBar](#)

[Installing ProgramBar](#)

[Using ProgramBar](#)

[Configuration of ProgramBar](#)

[Version release information](#)

[Wishlist of things to do](#)

[Known problems with ProgramBar](#)

[Toubleshooting problems in ProgramBar](#)

[Reporting bugs in ProgramBar](#)

[Shareware agreement and order form, statement of warranty,
contacting the author and other notes](#)

Introduction to ProgramBar

Windows was designed so that you might run several programs together. Using information from these programs so that you might do your job and hit that deadline. Reading a help file while trying to write and test that Excel based macro on the information requested in an e-mail for example.

While you can do this, you quickly run out of screen real estate in which to place all those open windows so you can see them all at the same time. Now you are having to switch between applications using Alt-TAB, more often than not overshooting the application you want - very annoying. ProgramBar will help you switch between applications quickly, easily and with confidence. It's as easy as using the mouse!

For most of the time you will not be aware that it is running on your system. It uses very little in the way of system resources and processor time as it sits in the background, quietly noticing when you load and close down programs.

The following main features have been implemented for this release:

- A 'Start' button that gives access to much of the functionality of ProgramBar.
- Access to, and running of, any program in your Program Manager groups.
- Access to all of the Control Panel configuration applets
- Each running application is represented on the ProgramBar as a button. A push of the button makes that application active.
- Right click on the ProgramBar panel to display a list of active applications that can be closed.
- Shutdown Windows, either returning to DOS, restarting Windows again, or rebooting your machine.
- Fast access menus for your most frequently used programs, documents and help files.
- Access to this help file (click on the ProgramBar logo on the right hand side of the bar.)
- Hidden out of the way when you don't need it. May also be permanently visible on your desktop.
- Fast search for files on your hard disk. They may be launched or added to the fast access menus.
- Run a program specifying the command arguments, or a data file with an association. The file can be browsed for.
- Colourful icons help navigation through the main popup menu.
- Flyby help hints when you let the mouse hover over part of the ProgramBar interface.

- Drag and drop of files from File Manager onto ProgramBar. Automatic separation into Applications, Documents and Help files.
- The state of ProgramBar is saved between sessions.
- The Control Panel may be inhibited from running in order that a potential clash between ProgramBar and Control Panel is avoided.
- Full configuration of ProgramBar via a popup dialog box.
- Applications can be excluded from the front panel task switching buttons to reduce clogging up of the panel.
- Task switching buttons on the front panel may now be stacked onto more than one row.
- ProgramBar may be triggered from any of the four edges of the screen.
- Applications/documents and help files added to the fast access menus may be edited to give more appropriate titles, run time arguments, or different working directories.
- Task switching now handles modal dialog boxes opened by an application better.
- Better support for people who use cursor screen wrap: the size of the region that can be triggered may be adjusted, the time delay before ProgramBar hides again may be increased.
- Menu and front panel fonts may be changed from within ProgramBar. A sample box also shows the suitability of the font for reproducing non-standard characters that may be used in titles.
- Clock and date displayed on the front panel.
- 3D effect on all dialog boxes though the automatic use of CTL3DV2.DLL if present. Tabbed dialog boxes adjusted to look more like their Windows 95 counterparts when used in conjunction with this DLL.

Installing ProgramBar

Using ProgramBar

Using ProgramBar

Activating ProgramBar

ProgramBar runs in one of two modes, permanently visible or hidden. When permanently visible the ProgramBar will never take focus from the application you are currently working with. When hiding (the default method of operation) the bar is not visible and requires the use of the mouse to activate it. This is the preferred mode of operation as ProgramBar will not then obscure status information that a maximised application might display at the bottom of the screen.

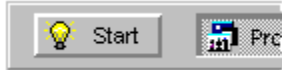
Just move the mouse pointer to a pre-defined edge of the screen and ProgramBar will pop up. It appears as a rectangular bar along the edge, the height and width determined by the triggering edge and the amount of information presented. Notice that the application currently running does not lose focus. When in this hiding mode, ProgramBar will automatically hide itself again a few moments after the cursor is moved off of its interface panel.

Known problems with ProgramBar

The interface components of ProgramBar

Three components of the interface are visible when activated. You can click on the visual components for more information, appearing as either a popup menu or a link to another page. Any jump links are also indicated in the body of the descriptive text.

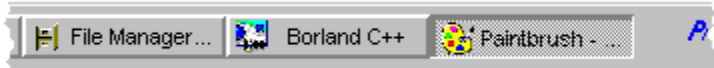
The Start button



This button on the left hand side of the bar gives access to much of the functionality of ProgramBar. Through it you may access the items in your Program Manager groups and load them. You can also configure any part of Windows that may be changed through the Control Panel application.

Press the button down to display [the Start menu](#). Drag the mouse up/down the menu to the item you want to activate. If an item will presents you with a sub menu then it will display an arrowhead on the right hand side. The sub menu will pop up when this item is selected, move the cursor off the left or right edge of the menu you are on and continue dragging the mouse up/down to continue selection. Some menu items may be more than one popup menu deep.

Running tasks



These buttons indicate the applications currently loaded on your system and detected by ProgramBar. The application that currently has the input focus is indicated by having its button permanently pressed in and shaded out. Each button has a brief description of the application derived from its title. These buttons always maintain the same relative positioning, with the most recently loaded program on the rightmost button.

Simply press the button for the application that you want to switch to. It will come to the foreground of the screen ready for you to use. The ProgramBar will automatically hide itself if it was hiding before it was used.

More information on tasks



When the cursor is held motionless over one of the task switching buttons more information is displayed. It appears in the form of a flyby 'hint' and gives the full title of the application window. This is useful should you have several instances of the same program and, because the button bar is crowded, names have been truncated so you can't distinguish between them.

Logo



More than just a pretty picture, the logo is also a button that may be pressed to activate this help file.

System menu

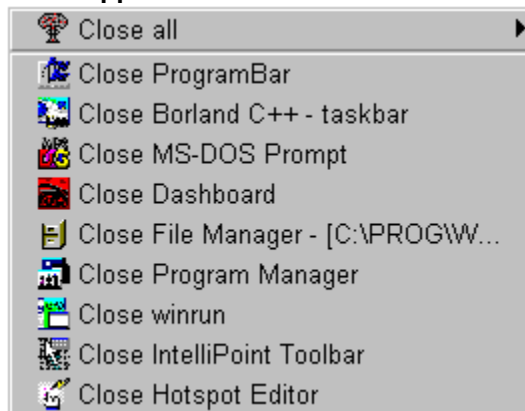


A single click with the left mouse button on a part of the ProgramBar panel that is not a button will display the system menu. From this menu you may close ProgramBar, or set whether or not it is always visible at the bottom of the screen.

You can also hide the ProgramBar. This is particularly useful if you normally have the ProgramBar permanently visible but want to see under the front panel for some period. ProgramBar will hide itself and pop up again when activated by moving the mouse to an activating edge. This is a one off hide and does not affect the 'always visible' state of the bar.

You may also turn the flyby hints on or off from this menu (they do annoy some people!)

Close Applications menu



A click of the right mouse button on a part of the ProgramBar panel that is not a button will display a menu that allows you to close any application running on the system. Keep the right mouse button pressed down as you drag the mouse pointer to select the required menu option. From this menu you

can also close all applications that are either minimized, maximized or running in a window smaller than the full screen.

ProgramBar will also terminate a DOS application running in a DOS-box. There will be no warning from the running program should it require saving of data. The shutdown is equivalent to pressing the Terminate button presented when the DOS-box system menu item *Settings...* is selected.
Use with extreme caution!

Activate the Borland C++ application

You just activated the application with the title 'Borland C++ - taskbar'. That application would have moved to the top of all of the windows and now have the input focus.

Activate the File Manager application

You just activated the application with the title 'File Manager'. That application would have moved to the top of all of the windows and now have the input focus.

Currently active application

The button for this application is pressed in and greyed out, indicating that this is the active application and already has input focus.

ProgramBar logo

When pressed the ProgramBar logo runs this help file.

Hide

May be used to hide the ProgramBar when it is permanently visible. Useful when the lower portion of the screen is displaying information that you want to see. The ProgramBar may be reactivated when the cursor is moved to the bottom of the screen.

Always Visible

This menu item toggles ProgramBar between being permanently visible and hiding when the mouse is moved off of the front panel. ProgramBar is always the topmost window, so it may obscure some information presented in the program you are currently working with.

Inhibit Control Panel

This menu item toggles ProgramBar between allowing and preventing Control Panel from executing. When checked, Control Panel is not allowed to load and, if already active when ProgramBar is first run, will be shutdown.

This option is provided so that the user may not accidentally have the same configuration windows opened by both Control Panel and ProgramBar. Since ProgramBar opens and controls a configuration window independently of Control Panel, having both open may cause unexpected side effects.

Show flyby hints

This menu item enables or disables the display of flyby hints. The hints assist the user in using ProgramBar, though some may consider them an annoyance.

When enabled and the mouse pointer has settled on a part of the ProgramBar interface for a few moments a small window appears. This window gives more information about where the mouse pointer has settled.

The most useful aspect of this at present is for the task switching buttons. Their flyby hint gives the full title of the window that will be selected. Normal operation of these buttons may truncate the title and make identification of similar programs difficult.

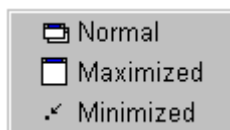
About...

Gives some information about the program and author.

Close ProgramBar

Shuts down ProgramBar. Note that Alt-F4 does not work with ProgramBar since it never has the input focus.

Close All menu



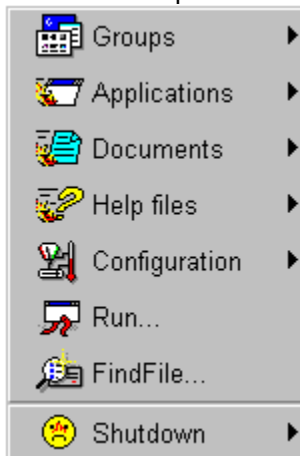
Windows grouped by type may be shut down together using this menu.

Close a program

Selecting an item here closes the application.

The Start menu

When pressed a menu pops up so that you can access the extra functionality of ProgramBar.



Click on the menu item that you want to learn more about. Use the back arrow to return to the

top of this page.

[Return to Using ProgramBar](#)



Groups

When selected a popup menu appears with a list of the program groups available in Program Manager. The contents of the list are sorted alphabetically. When an item on this list is selected then a further popup menu appears with a list of the icons present in that group. The title of the icon as stored by the Program Manager is used. If the number of icons in that group is too large then the list will be spread across two panels.

To run an icon, drag the mouse to the menu item and release the mouse button. If the menu item is a program then the program will run. If the menu item is a file then the program associated with the files extension will run and attempt to load the document.



Applications

When selected a popup menu appears with a list of applications that have been set up by the user for quick access. It is suggested that you place all the major applications that you use here so that they can be accessed more quickly than through the Groups menu item.

Applications may be added to this menu by selecting the executable from File Manager and dragging it onto the ProgramBar front panel, or creating a new item from [the configuration dialog box](#). If the executable is recognised (i.e. appears in one of your Program Manager groups) then it's icon title will be displayed, otherwise the full path to the executable will be shown.

Multiple files may be dragged onto ProgramBar. They will be separated and placed on the appropriate menu (Applications, Documents or Help files). If the executable is already on the menu then a duplicate will be added, on the basis that you may wish to edit the second executable to run with different command line options.

Applications may be deleted by use of the configuration dialog box.



Documents

When selected a popup menu appears with a list of documents that have been set up by the user for quick access. It is suggested that you place all the frequently accessed documents that you use here so that they might be accessed more quickly than through the Groups menu item.

Documents may be added to this menu by selecting the document from File Manager and dragging it onto the ProgramBar front panel, or using the [configuration dialog box](#). If the document is recognised (i.e. appears in one of your Program Manager groups) then it's icon title will be displayed, otherwise the full path to the document will be shown.

Multiple files may be dragged onto ProgramBar. They will be separated and placed on the appropriate menu (Applications, Documents or Help files). If the document is already on the menu then a duplicate will be added.

Documents may be deleted by use of the configuration dialog box.



Help files

When selected a popup menu appears with a list of help files that have been set up by the user for quick access. It is suggested that you place all the frequently accessed help files that you use here so that they might be accessed more quickly than through the Groups menu item.

Help files may be added to this menu by selecting the help file(s) from File Manager and dragging it onto the ProgramBar front panel, or using the [configuration dialog box](#). If the help file is recognised (i.e. appears in one of your Program Manager groups) then it's icon title will be displayed, otherwise the full path to the help file will be shown.

Multiple files may be dragged onto ProgramBar. They will be separated and placed on the appropriate menu (Applications, Documents or Help files). If the help file is already on the menu then a duplicate will be added.

Help files may be deleted by use of the configuration dialog box.



Configuration

This menu provides access to the [ProgramBar configuration dialog](#)., and the configuration utilities managed by Control Panel.

The ProgramBar configuration dialog is modeless, you can continue to use ProgramBar while it is active. In most cases, changes made to the configuration dialog will be immediately apparent in the operation or appearance of ProgramBar.

The dialog box associated with the Control Panel applet will appear in the bottom left corner of the screen. You will have to close or cancel the dialog box before control is returned to ProgramBar.

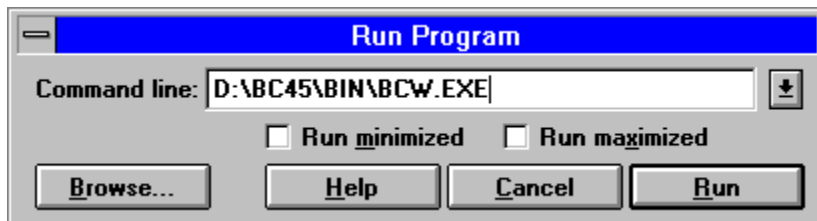
Some dialog boxes may sit partially off the edge of the screen. Just pick up the dialog box by the title bar and drag it so it is visible in entirety. Since the ProgramBar is the parent of the configuration dialog boxes then a loss of focus from the currently running task to the dialog box is inevitable.

***** WARNING *****

It is not recommended that the configuration utilities are run from both Control Panel and ProgramBar. Two dialog boxes will appear, and since they are being generated by the same DLL the results of any operation are uncertain and potentially fatal. It is possible to [inhibit Control Panel](#) while ProgramBar is in operation.



Run...



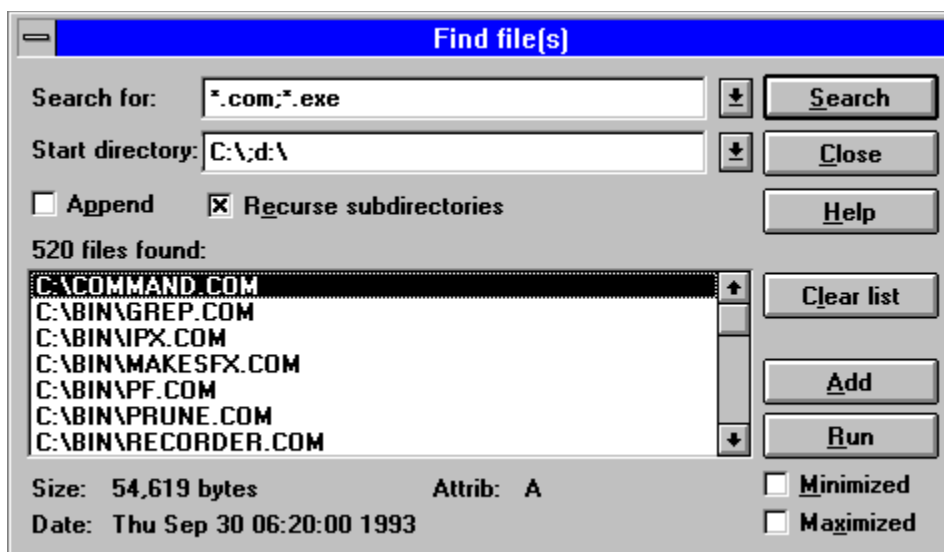
This dialog box allows you to find and run an individual program. It is also possible to run the program either as an icon minimized on the desktop or as a fullscreen application. The full path and filename of the executable should be typed into the combo box. A history list of up to 20 of the previous commands issued is available. The command line arguments should also be entered. Alternatively the name of a data file may be entered. If the data file is associated with an application then the application will be run and the data file loaded.

The browse button allows you to search for an executable or data file with the assistance of a dialog box. By default the browse dialog box searches for executables (*.exe,*.com,*.pif,*.bat), but any extension may be specified. It is not possible to set the arguments passed to the selected executable using the browse dialog box.

Click on the OK button to run the program/data file, or Cancel to close the dialog box without attempting to run the program. Should the executable not exist or the data file selected not have an association, no warning message will be generated.



Find file...



This dialog box helps you to find files by specifying the search pattern and search directory. The matching files may then be added to the fast access menus for executables, documents or help files, or executed.

Enter the search pattern into the 'Search' combo box. Typical patterns may search for all instances of one extension (e.g. *.com), one type of main filename (e.g. progbar.*), or for missing characters (e.g. abc??*.wri). Any search pattern accepted by the MS-DOS dir command may be used here. The 20 most recently used search patterns are stored in the combo box for recall. You may enter multiple patterns by using a semicolon (;) separator (e.g. *.com;*.exe).

Then enter the start directory that the search will begin with. By default directories are searched recursively. Clearing the 'Recurse sub-directories' checkbox modifies the search to consider only the indicated directories. For each new search the contents of the Files found list box are removed, unless the 'Append' check box has been selected.

The 'Files found' list box gives a list of the files that match the specified search pattern. They are sorted alphabetically by directory and alphabetically by filename within each directory. The list box also displays how many files it has found. The list box may be cleared using the 'Clear list' button. Appended files always appear at the end of the existing list, also sorted alphabetically by directory and filename.

By clicking on each of the files, the attributes of that file are displayed at the bottom of the dialog box. Groups of files may be selected by clicking and dragging the mouse in the list box. Individual files may be selected/unselected by clicking on the file while the CTRL button is held down. When more than one file is selected it is not possible to display file attributes.

Two actions may be performed on the selected files. First, they may be added to the fast access menus for applications, documents or help files by clicking on the 'Add' button. Alternatively they may be executed by selecting the 'Run' button. If the file is not an executable, or does not have an association known by File Manager, then the file will not be run or added to the fast access menus. Such files are quietly ignored, no warning message will appear. By selecting the 'Run minimized' check box then all the programs run will be iconized. Alternatively, the 'Run maximized' check box may be selected so that all the applications are run fullscreen. These check boxes only affect programs that you try to run from this dialog box.

Press the 'Close' button to close the dialog box and return to the normal operation of ProgramBar.

***** Warning *****

Windows 3.1 does not handle resources as well as Windows95, so selecting and running 100 files may well cause you system resource problems. Try it... once :)



Shutdown...

This menu item produces a popup menu with two options, one referring to Windows and the other to ProgramBar.

Choosing the Windows menu item allows you to close down Windows. There are three ways of doing this: a normal shutdown that returns you to the DOS command prompt; a reboot of Windows that may be used to introduce changes that cannot be changed on the fly (e.g. screen driver resolution); and a shutdown that performs a soft reboot, running through the POST and reloading of DOS. Each application is allowed to prompt you in case it has unsaved data, possibly allowing the shutdown to be aborted. Note that at the present it is not possible to perform a shutdown with a DOS session active.

Choosing the ProgramBar menu item will force ProgramBar to go through its normal shutdown procedure. It is equivalent to choosing `Close ProgramBar' from the system menu.

POST

An acronym for Power On Self Test. Performed by your computer when it is turned on, this checks the RAM of your machine, the presence of the keyboard, PS/2 mouse and other key components required for the normal operation of the machine.

ProgramBar configuration

Not yet implemented.

Command line

Enter the name of the executable and path (if required), and any additional arguments required. Alternatively the name and path of a data file that has an association known to File Manager.

The last 20 entries may be accessed using the button on the right of the combo box.

Run minimized

When selected the program will be loaded and run in icon form.

Run maximized

When selected the program will be loaded and run fullscreen.

Run

Attempts to run the program and arguments entered into the Execute combo box. If the program or data file does not exist or has no association then the dialog box will close with no error message.

Cancel

Cancels the dialog box. No program will be run and the contents of the Execute combo box will not be added to the history list.

Help

Gives more detailed information about how to use the dialog box.

Browse

Creates a dialog box that allows you to search for an executable or data file.

Search for

Enter the search pattern for the file(s) you want to find. Any search pattern accepted by the MS-DOS dir command is acceptable. A '*' represents any number of characters, a '?' represents a single character. e.g. *.com, progba?.*

Use a semicolon (;) to separate multiple search patterns.

Start directory

Specifies the directory that the search is to begin from. Must be a valid directory. Use semicolons (;) to separate multiple search directories.

Append

When checked the results of the next search are added to the contents of the 'Files found' list box. When not checked the listed files are deleted from the list box before the search is performed.

Recurse sub directories

When checked all directories below the start directory will also be examined for files matching the search parameters. When not checked only the specified directory will be searched.

Search

Initiates the search with the supplied parameters.

Close

Closes the dialog box. All search results will be lost, but the 20 most recent search parameters will be retained in each of the 'Search for' and 'Start directory' combo boxes.

Help

Gives more detailed information about how to use the dialog box.

File count

Indicates how many files have been found that match the specified search parameters.

Files found

This list box contains all the files found using the specified search parameters. Selecting a single

file gives information on the file: size, date/time stamp and attribute flags. Multiple files may be selected by clicking and dragging over a range of files. The list box will scroll as required. Individual files may be selected/de selected by clicking on them while holding down the CTRL key.

File attributes

Displays the attributes of a single file. The file length, date/time stamp and access attributes are given.

The access attributes are as follows: R - read only, A - archive bit set, S - system file, H - hidden file.

Clear list

Forces the contents of the 'Files found' list box to be emptied. Enabled only when there are entries that can be flushed.

Add

Adds the selected file(s) to the fast access menus. Enabled only when a selection has been made. Files that do not have associations known by File Manager will not be added.

Run

The selected files will be launched. Any file that does not have an association known to File Manager will not be run.

Run minimized

When checked, programs that are run by selecting the 'Run' button will run as icons. This button does not affect how items are added to the fast access menus using the 'Add' button.

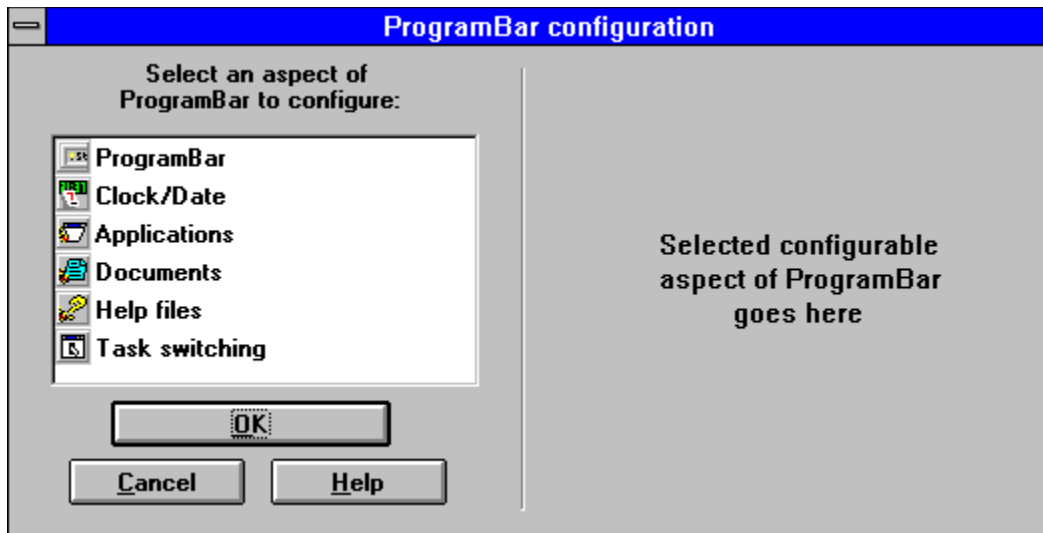
Run maximized

When checked, programs that are run by selecting the 'Run' button will run as maximized windows. This button does not affect how items are added to the fast access menus using the 'Add' button.

Control Panel items

Gives a list of the configuration utilities managed by Control Panel. Select the menu item for the configuration utility that you want to access.

Configuration of ProgramBar



The configuration menu may be accessed by selecting Configuration|ProgramBar from the Start menu. A dialog box appears, split into two parts. The left hand side present a list of the aspects of ProgramBar that can be configured by the user. Selecting one of the items on the list will display an appropriate panel on the right hand side of the dialog box.

The following sections of ProgramBar may be configured:

[ProgramBar - Main](#)
[ProgramBar - Fonts](#)
[Clock/Date](#)
[Applications](#)
[Documents](#)
[Help files](#)
[Task switching](#)

Editing the INI file

Previous versions of ProgramBar required tinkering in `progbar.ini` in order to gain some benefits. This required some documentation of the structure of this file. From this version onward all changes to `progbar.ini` can now be made through ProgramBar itself, so the `ini` file will not be documented in this and future releases. Tinkering with the `ini` file is not recommended for the faint hearted!

[Return to the contents page](#)

OK

Accepts the changes made to ProgramBar and commits them to the INI file.

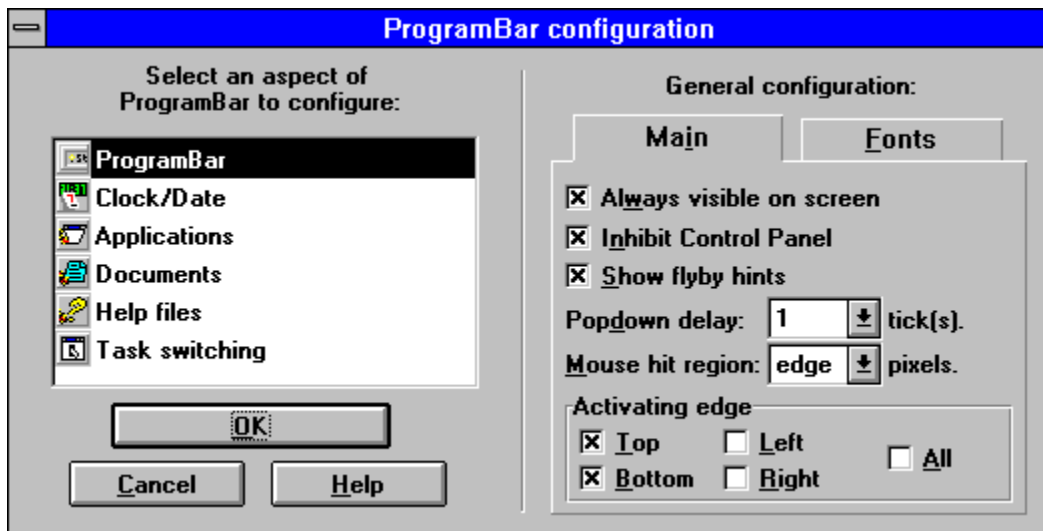
Cancel

Aborts the changes made in the configuration dialog box and returns ProgramBar to the state it was in before the Configuration dialog was opened.

Help

Runs this help file, jumping to the main configuration help page.

ProgramBar configuration - Main



Click on a dialog control for more information.

This configuration panel allows you to change the default behaviour of ProgramBar. This includes how you trigger ProgramBar and the parameters that control how 'sensitive' ProgramBar is to your actions. You may also select the edge(s) that ProgramBar may be triggered from for your convenience.

Always visible on screen

When selected the ProgramBar front panel will always be visible on the screen at the last edge it was triggered from.

Inhibit Control Panel

When selected Control Panel is prevented from loading. Included as a preventative measure so that a potential clash between ProgramBar and Control Panel is avoided.

Show flyby hints

Allows small hint windows giving more information when the mouse hovers over a control on the front panel.

Popdown delay

Adjusts the time delay between the mouse moving off of the front panel and ProgramBar hiding itself. Each tick is approximately half a second. This parameter only affects ProgramBar if the 'Always visible' check box is not set.

Mouse hit region

Specifies how close the pointer has to be to the edge of the screen before ProgramBar pops up from hiding.

Top edge activation

If selected ProgramBar may be activated from the top edge of the screen.

Bottom edge activation

If selected ProgramBar may be activated from the bottom edge of the screen.

Left edge activation

If selected ProgramBar may be activated from the left edge of the screen.

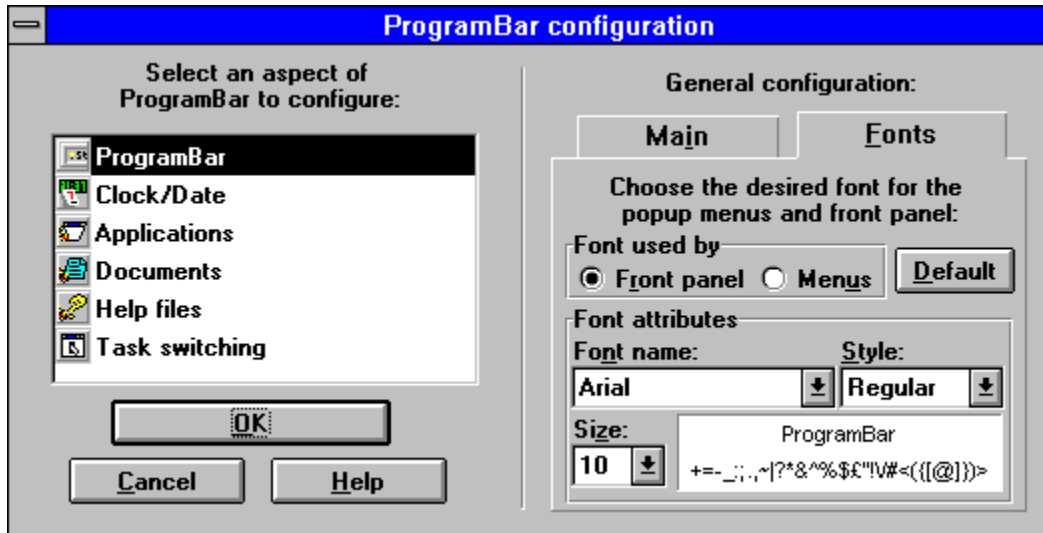
Right edge activation

If selected ProgramBar may be activated from the right edge of the screen.

All edges activation

All edges of the screen may be used to activate ProgramBar.

ProgramBar configuration - Fonts



Click on a dialog control for more information.

This dialog helps you select the font that may be used on the ProgramBar front panel and in the ProgramBar popup menus. A preview of the font is displayed, along with a range of characters that might be seen in a Windows title. Should some of these characters appear as squares then the selected font does not support that character. This may affect your choice of font.

Front panel font

When selected the attributes of the font used on the front panel are made available for editing.

Menu font

When selected the attributes of the font used on the main and system menus are made available for editing.

Default font style

When pressed this button restores the selected font to its default value or Arial 10 point.

Font name

This is the typeface family name of the font. ProgramBar will enumerate all of the fonts available on your computer and display the appropriate sizes and styles available.

Font style

This is the style in which the font will be displayed. For non-True Type fonts either Regular or Bold will be available at a minimum. For True Type fonts only the styles supported on your machine will be available.

Font point size

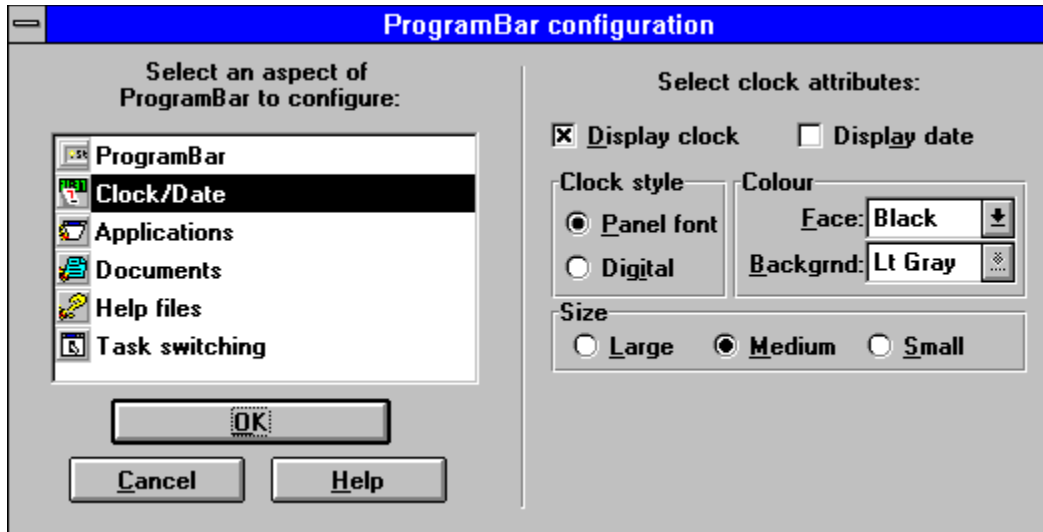
The range of point sizes accepted lie between 7 and 14. A True Type font may be displayed in all point sizes, while a non-True Type font will only have a fixed subset of this range. At small point sizes a font from a different family may be substituted in order to make the display more readable. This is beyond the control of ProgramBar.

Font sample

A sample of the font size and style are presented so that the user may make a more informed choice about an appropriate font.

Many fonts are incomplete in their representation of all the available characters, so a subset of the more unusual characters is also displayed. If any of these characters appear as a square then the selected font does not provide a definition for that character. This may affect the appearance of the front panel if the character is used in the title of a window.

Clock/Date configuration



Click on a dialog control for more information.

The type of clock displayed on the front panel may be selected from this dialog. If the panel font is used then the short date may also be displayed. The time and date format are obtained from the settings in Start|Configure|Control Panel|International. Changing the 12/24-hour format there will automatically be reflected in the appearance of the clock. If the cursor is allowed to hover over the clock then the flyby hint displayed will give the long date.

Display clock

When the check box is selected the clock appears on the front panel of ProgramBar.

Display date

When the panel font is used the current date may be viewed when this check box is selected. The date appears in the short format defined in the Start|Configuration|Control Panel|International dialog box. It is not possible to display the date when the clock uses a digital font.

Use panel font

The same font used on the front panel is also used to display the clock (and date if selected). The time format used (12/24 hour, display leading zero etc.) may be set using Start|Configuration|Control Panel|International.

Use digital clock face

The numbers are displayed using a digital font. The font is available in three sizes. When a twelve hour format is selected in the Start|Configuration|Control Panel|International dialog box the morning is indicated by a mark above the separating colon (:). The afternoon is indicated by a mark below the colon.

Clock face colour

The text colour of the characters used to display the time may be set here.

Clock background colour

The background colour of the digital clock may be set here. It is not possible to change the background colour when the panel font is used.

Large digital clock face

The digits used to display the clock are the largest possible. This may be useful for higher resolution displays.

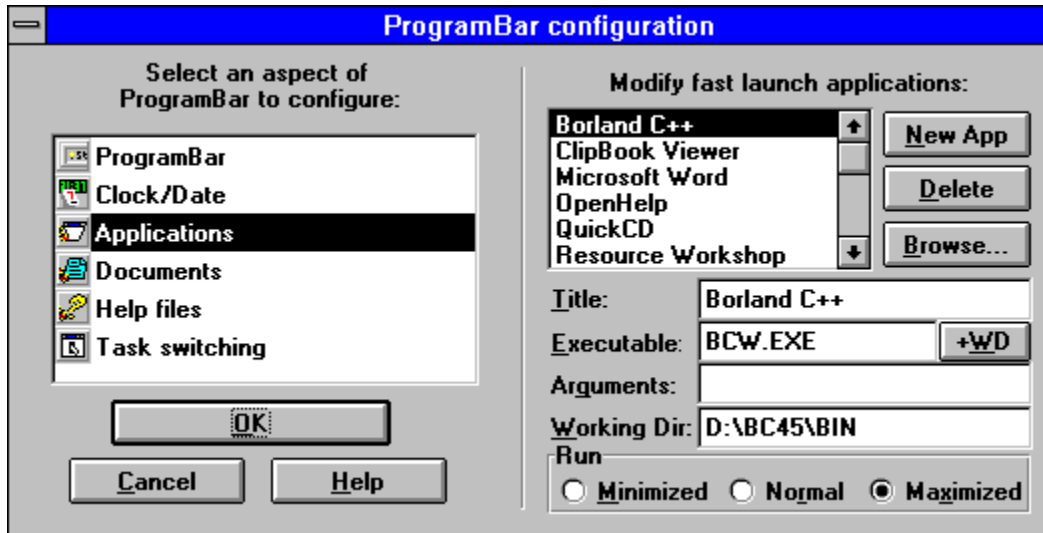
Medium digital clock

The digits used to display the clock are intermediate in size. Most useful for the lower sized SVGA modes like 800x600.

Small digital clock

The digits used to display the clock are as small as possible. Most useful for the lowest screen resolutions, 640x480 and below.

Applications configuration



Click on a dialog control for more information.

From this dialog it is possible to change the attributes of the entries on the Application fast access menus. Selecting an item in the list box brings up the full details of that item, including its title on the fast access menu, the executable name, the arguments that the program is run with and the working directory for the application. You may also set whether the application is run full screen, minimized or at its default desktop size.

The application executable may appear without its full directory path. In this case the application executable must be on the path if no working directory is specified. If a working directory is specified then the path and the working directory will be searched for the executable. A full path may be specified, in which case the working directory is where the program will look to load files from by default. Some applications may override this once loaded.

You may drag executables from File Manager into the ProgramBar front panel, they will be inserted into the list box if the configuration dialog box is open. Title changes to entries in this dialog are immediately effective on the fast access menus, changes to the executable/working directory/arguments are effective after changing items in the list box, or moving on to select another configuration dialog.

*** Important ***

If the executable file is not on the path, in the Windows directory or Windows system directory, and has no path prepended to it then changing the working directory will prevent the program from being found. The +WD button may be used to prepend the working directory to the executable file before the working directory is modified. You will be warned if the executable cannot be found when the dialog is closed.

List of applications

This list box presents a list of items that are currently on the applications fast access menu. They are sorted alphabetically. Select an item on the list to view its details in the rest of the dialog.

New application

Add a new application to the fast access menu.

Delete application

Removes the application from the fast access menu.

Browse applications

Opens a common dialog box that assists in the search for applications. It is not possible to enter a file without one of the extensions .EXE, .COM, .BAT or .PIF. A message box informs the user of this if the attempt is made.

Title of application

The title of the application appears on the fast access menu and in the applications list box. It should be used to uniquely identify the item.

Executable file

This field contains the name of the executable file searched for on the harddisk. If no path name is prepended then the working directory, Windows directory, Windows system directory and DOS path will be searched for the executable. If a path name is prepended then only that directory will be searched.

Prepend working directory

This button prepends the current working directory to the executable filename. If a full path is already present then confirmation is requested.

Should you wish to change the working directory and you *know* that the application cannot be found in the Windows directory, Windows system directory or on the DOS path then you must prepend the working directory to the executable filename in order that the executable may be found.

Arguments to the executable

Specifies additional command line arguments that are to be passed to the application.

Working directory

Specifies the default directory in which a program will try to load files from.

If the executable resides in this directory and the directory is not the Windows directory, Windows system directory, or on the DOS path then changing the working directory means the executable will not be found. Use the **+WD** button to prepend the working directory to the executable to avoid this problem.

Run application minimized

When launched the application will appear as an icon on your desktop.

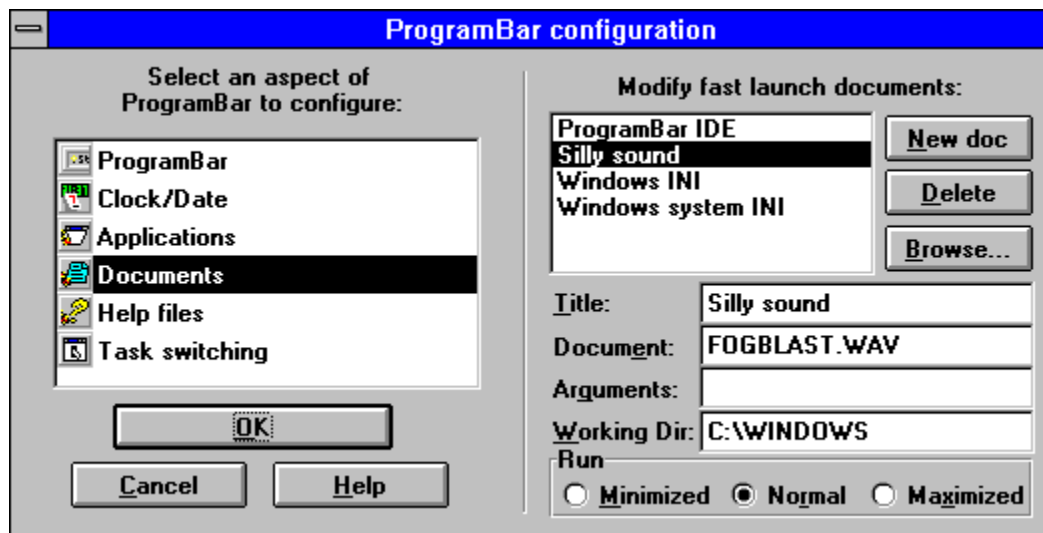
Run application normal sized

When launched the application will appear as a normal sized window on your desktop.

Run application maximized

When launched the application will appear full screen on your desktop, obscuring all other running applications.

Document configuration



Click on a dialog control for more information.

From this dialog box it is possible to change the attributes of the Documents on the fast access menu. Selecting an item in the listbox shows its details. You may select additional arguments to add to the running of the application associated with the document.

The path to the document and the file are stored separately in the document program name and working directory respectively. This separation is notationally more convenient for the running the application associated with the document. The application will use the working directory as the place in which to find the document, and then as the first place it will look when the user wants to load a similar

document.

Document list

This list box contains a list of items that are currently on the document fast access menu. They are sorted alphabetically. Select an item to view its details.

New document

Add a new document to the fast access menu.

Delete document

Removes the document from the fast access menu.

Browse document files

Opens a common dialog box that assists in the search for document files. It is not possible to enter a file that might appear on the help of application fast access menus. A message box informs the user of this if the attempt is made.

Document title

The title of the document appears on the fast access menu and in the selection list box. It should be used to uniquely identify the document. If the document is not in one of the Program Manager groups then its full path and filename are displayed by default.

Document file

This field contains the name of the document that will be loaded into the associated application. A full path may be specified, if not then the document will be searched for in the working directory, the Windows directory, the Windows system directory and the DOS path.

Document arguments

Optional arguments that may be specified on the command line of the application.

Document working directory

The directory in which the document may be found if no path is prepended to the filename specified for the document file.

Run document minimized

The application associated with the document will be run minimized on the desktop.

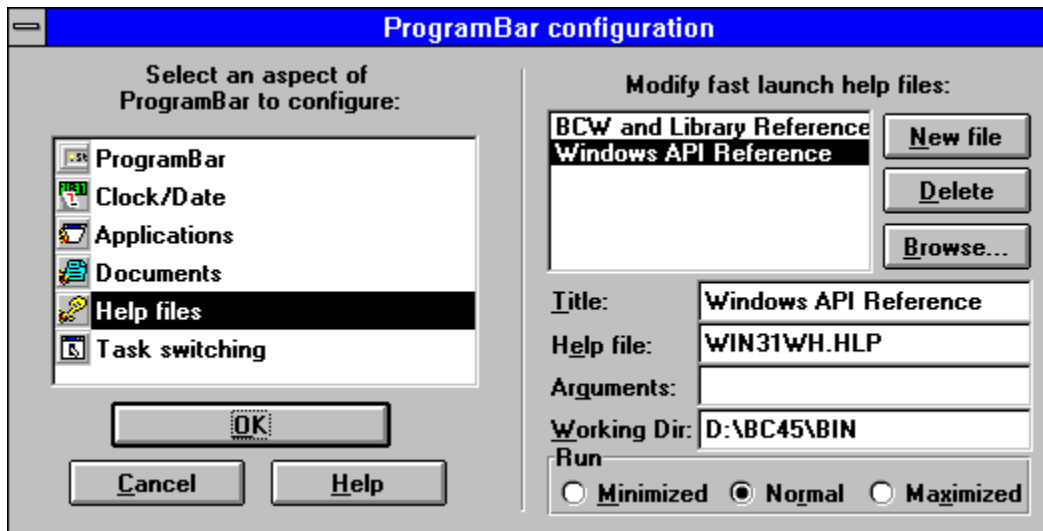
Run document normal

The application associated with the document will be run in a default sized window.

Run document maximized

The application associated with the document will be run in a maximized windows, obscuring all other windows on the desktop.

Help files configuration



Click on a dialog control for more information.

This dialog box may be used to change the attributes of the help files on the fast access menu. On selecting an item in the list box, its details will be shown. If the help file or working directory are changed then ProgramBar will ensure that the file exists before allowing another help file to be selected or another dialog entry to be chosen.

Help file list

This list box contains all the items that appear on the help file fast access menu. Select an item to view its details.

New help file

Adds a help file to the fast access menu.

Delete help file

Deletes a selected help file from the fast access menu.

Browse help files

Opens a common dialog box that assists in the search for help files. It is not possible to enter a file without the help file extension, a message box informs the user of this if the attempt is made.

Help file title

The title under which the help file is presented in the fast access menu and the help file list box.

Help file

Filename which contains the help file. By default this file is searched for in the working directory unless a full path is specified.

Help file arguments

Specifies addition arguments on the help file. Actually I'm not even sure that there can be additional arguments...

Help file working directory

Specifies the directory in which the help file will be found. If a full path is specified for the help file then this entry is redundant.

Run help file minimized

When run the help file is displayed as an icon on the desktop.

Run help file normal

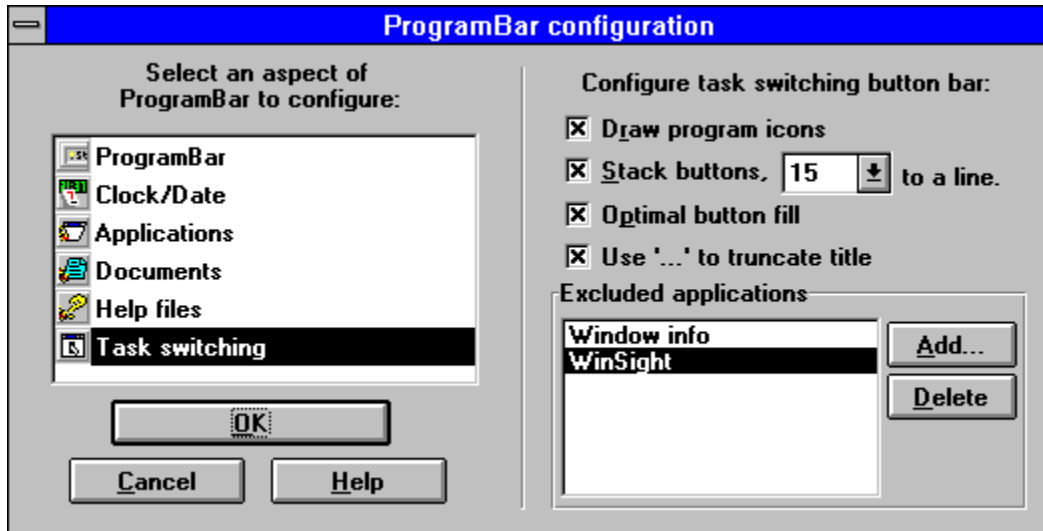
When run the help file is displayed in the default sized window. This will depend upon previous usage of the help application.

Run help file maximized

When run the help file is displayed in a maximized

window on the desktop.

Task switching configuration



Click on a dialog control for more information.

This dialog may be used to control the format in which the ProgramBar front panel displays active applications. For individuals who have a large number of applications running at any one time, this allows the front panel to display a significant portion of an applications title - making it easier to recognise applications.

It is also possible to exclude applications from the front panel. Many applications create top level windows that display useful information, however ProgramBar recognises them as a separate main window that you may want to switch to. This can clutter up ProgramBar unnecessarily. ProgramBar allows you to exclude these windows by trying to find unique details by which they may be recognised.

Draw icons

When this check box is set the button bar displays icons representative of the applications running on your desktop.

Stack icons

When selected the task switching buttons on the front panel will be spread onto more than one line when a predefined number of icons occupy one row. This option only affects ProgramBar when it has been triggered from the top and bottom edges.

Size of icon stack

This value determines the point at which ProgramBar will add an extra row to the task switch buttons. By adding this row ProgramBar will not become overly congested with buttons that obscure an applications title text.

Optimal fill

When checked this option allows ProgramBar to use the largest size of button when more than one row is required to display the available tasks. Selecting this option prevents the visually unsightly extra button or two hanging on at the start of a new row.

Use '...' to truncate title

When selected the window title displayed on the task switching button will show it has been truncated with ellipses (...). This takes up more space and allows less of the title to be displayed.

List of excluded applications

A list of application windows that have been excluded from the array of task switching buttons. The name of the application is derived from a window title, truncated if necessary. Use the **Add...** and **Delete** buttons to modify this list.

Add an application to the exclude list

This button presents the user with a list of application windows currently appearing on the array of task switching buttons on the front panel. The name of the application is truncated from the full window title.



By double clicking on an entry or using the **Detail...** button more information may be obtained on how the application is to be excluded. ProgramBar uses the class name of the window and the module name owning the window to exclude the window. The full title is provided for information purposes only.

Press the **Add** button to exclude the selected window.

Delete an application from the exclude list

An application window may be deleted from the exclusion list once it has been selected by pressing this button.

Installing ProgramBar

There are three files required for the correct use of ProgramBar. They are:

PROGBAR.EXE	The executable that provides all of the functionality of ProgramBar.
PBHOOK.DLL	A DLL that is used by ProgramBar to monitor system activity, notify of programs being opened or closed.
PROGBAR.HLP	This help file.

In addition, the following two files should also be present in the archive:

README.TXT	Text instructions on how to install the program plus additional information.
FILE_ID.DIZ	A file that assists BBS operators in maintaining their archives.

To install ProgramBar follow the steps below:

1. Create a new directory for ProgramBar. (e.g. C:\WINAPPS\PROGBAR)
2. Copy all the files listed above into this directory.
3. Then activate *Program Manager* and choose the program group that you want to place the ProgramBar launch icon into.
4. From Program Manager's menu chose *File|New...*

5. Select *Program Item*, then press the OK button.
6. Fill in the name of the icon and the path where the executable may be found.
7. If you want to have ProgramBar launch when you run Windows, place a copy of the icon in the *StartUp* group. This may be done by keeping the CTRL key pressed down while clicking on the ProgramBar icon and dragging it to the *StartUp* group.

Clicking on the ProgramBar icon will now run the program. ProgramBar creates an initialisation file called `progbar.ini` in the windows directory and a file in the same directory as its executable called `PROGBAR.DAT`.

ProgramBar assumes that both the Program Manager and WinHelp executables are available. Some default icons are taken from these files for display purposes.

[Introduction to ProgramBar](#)
[Return to the contents page](#)

Version release information

This page details the changes made to ProgramBar between version releases. Several of these changes have been as a result of the feedback from earlier versions, my thanks go out to all of you who took the time to write and show your appreciation of the effort I've spent on this program.

Version 2.11, 1-Dec-95

- ProgramBar failed to handle a large number of applications on the front panel, generating a 'TaskBar overloaded' message. This message should now only appear under extreme unreasonable circumstances (ie when the buttons are so narrow that they cannot display their contents.)
- Control Panel applets are now cached, significantly decreasing ProgramBar's load time. If any of the applets used by Control Panel are replaced, deleted or added to then ProgramBar will re-cache the data where required. Data is also re-cached if the screen driver is changed as the icon format may be different.
- Control Panel applets in the [MMCPL] section of `control.ini` are now scanned as well as the [drivers.desc] section.
- A peculiar window create/destroy/create cycle that assigned the same window handle in both cases resulted in two buttons being displayed on the ProgramBar front panel. E-mail Connection was the only application seen to exhibit this behaviour. Fixed.
- Added a switch that allows the display of ellipses (...) on the front panel task switch buttons to be turned off. Useful for when a large number of buttons are displayed.
- Confirmation of closure of a DOS box via right clicking on the front panel is now required. A strongly worded message reminds the user that resources may be lost and Windows may be impaired.
- The Find file dialog box now defaults to searching subdirectories.
- Most error dialog box and menu message strings moved to the resource file for easier internationalisation of the executable.
- Multiple arguments may now be specified for file wildcards and directories in the Find file dialog box. Each argument must be separated by a semicolon (;).
- Button text colours on the front panel are now fixed to black regardless of the choice of personal colour scheme.
- CTL3DV2.DLL now used for dialog boxes if it is installed. The 3D menus and front panel buttons are not and never have been provided using this DLL. Tab dialog boxes provide a Windows 95 look and feel when this DLL is detected.
- ProgramBar crash protection code improved, now distinguishes between modules of the same name.
- Annoying screen flicker caused by selecting a new dialog page on the

configuration dialog removed.

- Intermittant problem when switching from a DOS full screen app back to Windows caused the button indicating the active app to have a black background. Technique for drawing buttons changed, fixing problems.
- Split menus crashed ProgramBar when sorted after an insertion. Most likely to occur on the fast access menus. Fixed.

Version 2.1, 21-Nov-95

- Non-standard Windows colour schemes interfered with icon masking using monochrome bitmaps. Menu icons gained lurid clashing colour backgrounds. Fixed.
- If both clock and date were not displayed on startup of ProgramBar then a recursive loop was entered, locking Windows. Ctrl-Alt-Del aborted ProgramBar, but Windows was left in an unstable state and running ProgramBar again returned the user to the DOS prompt without warning. Fixed.
- Code has been installed to ensure that should ProgramBar crash for any reason, it is now safer to re-run the program. Due to the nature in which ProgramBar interacts with Windows it cannot be guaranteed to prevent further crashes/lockups/unexpected kickouts to DOS.
- Browse buttons added to fast access configuration dialogs to complement the drag/drop addition of files.

Version 2.0, released 18-Nov-95

- Added full configuration to ProgramBar via a popup dialog box. Most changes to the configuration dialog are immediately reflected in the appearance of ProgramBar. All changes can be easily cancelled.
- The Find File dialog box now provides a default search directory, it will be the root directory of the first fixed or remote disk on your system (most likely C:\).
- When using the Find File dialog box to append a second set of files, the first file to be appended was incorrectly placed somewhere in the middle of the first list. Fixed.
- Better handling of desktop applications that don't define class icons.
- Ability to exclude applications parent windows from the front panel task switching based on module name and window class. Included ability to preview the criterion used to eliminate the parent windows.
- Better display of front panel task switching buttons. They can now stack onto more than one row. User configurable.
- Full editing of applications on the fast access menus. Entries may now be run minimized/maximized/normal, arguments can be appended, working directory may be changed, title displayed on the fast access menu can be

edited. Settings saved to ini file are incompatible with v1.x, but upgrade performed automatically.

- While finding the control panel applets, entries in the `[drivers.desc]` section of `control.ini` were assumed to be in the Windows system directory. ProgramBar could not find drivers that included a full path in their entry. Fixed.
- Added a `/d` command line option to produce a debugging script during start up. Currently limited to examining loading of DLL's and opening files.
- When switching to an application using the button bar, if the application had been disabled by a dialog box then the focus was incorrectly set to the application (not the dialog box). ProgramBar now searches for the most recent dialog box that has disabled the application and sets the input focus there.
- ProgramBar may now be accessed from the left, right and top edges of the screen in addition to the bottom edge.
- The size of the region that can activate ProgramBar may now be modified. Added for the benefit of those who use cursor wrap and can't hit the edge pixels reliably.
- The time delay between moving the cursor off of the front panel and ProgramBar hiding itself may now be adjusted by the user.
- Choice of font used on the front panel and in menus moved from `progbar.ini` to the configuration dialog box. Dynamic adjustment shows what the new front panel will look like (if visible on the desktop) before the change is committed.
- Better handling of close down of Control Panel if it has a dialog box open. ProgramBar now attempts to close the dialog box before closing Control Panel.
- ProgramBar front panel now appears briefly on startup to indicate an edge that may be used to access the front panel from. For the first time user the main window appears permanently on by default, several people failed to find ProgramBar once it had finished displaying the logo screen.
- Enhanced detection of parent windows that may be switched to. ProgramBar failed to handle properly windows that had been hidden or had zero size. In several cases windows were not seen to close by ProgramBar and activating buttons were not removed from the front panel.
- Closing an application with a right click now attempts to close dialog boxes associated with the application first.
- Clock and date added to front panel. Either a digital or panel font clock may be displayed. If the clock uses the front panel font then the short date may also be displayed. The digital clock is not able to display the date. The fly-by hint associated with the clock displays the long date, as defined in `Configuration|Control Panel|International`.

- Corrected several omissions and factual errors from the help file, as well as writing up all the new features.
- Fixed a bug that crashed ProgramBar with a 'Divide by Zero' error if there were no programs on the desktop when ProgramBar was run.
- Reduced the chance of a text clipping problem occurring on the front panel buttons. Text clipping may still occur because of kerning by the GDI font driver.

Version 1.2, release 23-Oct-95

- Crashing bug. When a popup menu is closed and the cursor is at the very bottom edge of the screen, ProgramBar locks up. Fixed. This is the only improvement to this version, but warrants a new release.

Version 1.1, released 22-Oct-95

- Task switching to full screen DOS application from the button bar failed, doing nothing. Fixed.
- ProgramBar failed to popup if the cursor was at the bottom edge of the screen and sitting over an icon title or window resizing frame. Fixed.
- ProgramBar became confused if another application forced ProgramBar to display/hide its main window, refusing to popup when hidden. Fixed.
- An application that forced ProgramBar to minimize its main window succeeded. This should now not be possible.
- Cursor now changes to an hourglass while ProgramBar loads or performs a time consuming operation.
- All menus now display colourful icons to aid navigation.
- A Program Manager item that specified a working directory different to the full path prepended to the executable file name could not be run. Items that did not specify a directory in which to find the executable were unaffected. Fixed.
- The ProgramBar panel and menu display fonts and point sizes may now be adjusted separately in the `progbar.ini` file for those people using high resolution display modes.
- Flyby hints for the front panel give more information than the (possibly) truncated title text.
- While scanning for the DLL's and drivers managed by Control Panel, a non-existent driver will no longer produce a 'File not found' dialog box during start up.
- Menu handling improved, menu's now split when they are larger than the screen height rather than a fixed number of entries. The list of Program Manager groups will now split into two if you have an excessive number of groups. This is a hint that you should rationalise the number of groups in Program Manager :)

- Added Maximized check boxes to the file finder and program run dialog boxes.
- FILE_ID.DIZ added for the benefit of BBS Sysops.

Version 1.0, released 9-Oct-95

- First public release.

[Return to the contents page](#)

I use the directory **C:\WINAPPS** to place most of my windows programs in. That way I can keep them separate from my Windows installation directory (**C:\WINDOWS**) and not clog up that directory unnecessarily (Window does a good enough job as it is!).

Known and potential problems with ProgramBar

Most of the compatibility problems that I have encountered have been in trying to get ProgramBar to work with Borland Dashboard 2.0. I have detailed below some of the compatibility issues I have successfully and unsuccessfully redressed. The other main source of problems lies in identifying a program that has just been launched so that the task switching buttons can be correctly updated.

[Using ProgramBar](#)

[Return to the contents page](#)

[Reporting bugs in ProgramBar](#)

Identifying applications appropriate for task switching

ProgramBar attempts to identify the tasks on your system that are to be considered applications. Anyone who has looked at a Windows resource monitoring program such as Winsight (Borland) will now that it is not always easy to separate the wheat (applications) from the chaff (all the other windows in the system).

When I first tackled this problem I was surprised at the different number of ways that an application may place a top level window on the screen. Most of the time I work with a small subset of the large number of applications I have stored on my hard-disk. Quite obviously I can't test all of the applications out there for compatibility. After some testing I have settled on the following strategy for determining if a window may be considered an application:

- If a window is hidden, it cannot be an application.
- If the window doesn't have one of the following it cannot be an application: A minimize box, a maximize box, a thick (re-sizing) frame.
- If it has a parent window, then it cannot be an application. An exception to this is if it has a parent window that has zero size.
- If the top level window has zero size then it cannot be an application. This is new to v2.0

This manages to exclude applications that only have an icon presence on the desktop, I felt that these programs would only clutter up the active window bar unnecessarily.

Note that an application may place several top level windows on your desktop, ProgramBar treats each of these windows as a separate 'application' and will create a button on the front panel for each such window.

If you come across a program that behaves oddly given the above criterion and can analyse why then I am interested in hearing from you so that I can update ProgramBar appropriately.

Closing down DOS sessions

ProgramBar can only close down a DOS session if TerminateApp is used. This is undesirable

since some system resources (notably DOS memory and file handles) may be lost. In addition, the shutdown of Windows is prevented by the presence of DOS sessions. I'm working to try and find a way around these limitations.

Finding the icons used by an application

When a Windows application is run it registers a class for each type of window. Each class includes within it a default icon that can be used when a window of that class is created. This default icon is available by inspection of the class and is used by ProgramBar to identify the application. However it may be overridden in two ways, both of which suggest that ProgramBar is using the wrong icon when in fact it isn't.

The first place where this icon *appears* to be overridden is in Program Manager. This is just the icon used to identify the program in the group, this information is not passed to the program when it is executed. By default Program Manager uses the first icon in the file. If the program registers an icon in its window class at startup, ProgramBar will use this icon rather than the one used by Program Manager.

The second way that a window can override its icon is by specifying a NULL icon in its class. In this case, every time Windows needs to draw the icon, it sends a request to the application to draw the icon for its window. The application is now free to respond by drawing directly into the region either text, graphics, or an icon it has stored in its resources. It is not possible (to my knowledge) to intercept this in a simple way. When confronted with this case ProgramBar will use the first icon in the programs resources, or failing that a default icon from `MOREICONS.DLL`.

Borland Dashboard 2.0

Supplied with the installation CD-ROM for Borland C++ 4.5, this application (now with Starfish Software?) gives you quick access to Program Manager groups, manages printing, displays system resources, allows an extended desktop, quick launch for applications and more. I use it mostly for the extended desktop and quick launch. ProgramBar might be seen by some to supplant some of the functionality of Dashboard. It's certainly been a minor headache trying to get the two to co-exist reasonably peaceably...

- Dashboard allows access to extended screens through either the program interface or hot keys. If this is done while ProgramBar is active then Dashboard will attempt to move the interface window off screen. ProgramBar intercepts this and prevents it occurring without having to be registered with Dashboard as being a 'sticky app'. This locking down of ProgramBar is seamless and does not require a screen redraw.
- The extended screens offered by Dashboard leave several unresolved problems. On several occasions I have noticed that when an application on a different extended screen is switched to the main display also fails to move to the extended window so that the active app can be seen. I have also seen the problem with the use of fast Alt-TAB so I suspect (but cannot prove) that the problem lies with Dashboard.
- Another problem with the extended screens lies in the activation of iconized programs. When a program that is minimized in a different extended window is activated, then the program is restored to the current active window. This problem is apparent for some applications and not others. I suspect this has something to do with the way ProgramBar is activating applications. I'm working on a solution to this.

Other possible problems I cannot test for

Many graphics cards these days allow for a virtual desktop that is larger than the physically displayed screen area. Since I am developing on a 3 year old 486DX/33, my graphics card doesn't support this feature (surprise!) Thus I have no way of testing how well ProgramBar works in these environments. It will either 'float' at the position on screen where it is first instantiated, or only popup when the lowest line of the virtual desktop is at the bottom of the screen. Some feedback on this would be appreciated so I can improve ProgramBar so it works on the widest possible range of machines.

Shareware, some legalese, and other notes

Shareware is a method by which software may be evaluated by the end user before purchase. Almost all of the shrink-wrapped software produced commercially in large volumes (and at comparatively high prices!) provide their installation disks in licensed and sealed packs. On opening the sealed packs the user is no longer allowed to return the software and obtain a refund - clearly you cannot find out if a software package will do what you need without committing a significant amount of money beforehand.

With shareware the end user can evaluate the software and decide if it benefits their working practices. If it doesn't then the software can just be deleted from their hard-disk. If, however, the software is found to be useful then the user is expected to pay for it. Just like shrink-wrapped software.

Your continued support of shareware gives the programmer the incentive to continue developing and improving the software. Shareware means distribute the program freely, not use the program freely.

ProgramBar is shareware. You have 14 days in which you may evaluate the software before you have to pay for it or delete it from your machine. Once paid for all subsequent upgrades may be used for FREE! All users who showed their support for version 1.x of ProgramBar under its bookware status by sending either a book or a postcard are not required or expected to re-register.

This software has not been crippled, nor has it any timer expiry. It is not my wish to have to add code like this to ProgramBar, I'd much rather be putting in new features for people to use. However, since shareware requires the trust of the user to pay for the product or delete it from their machine then I may be forced into adding a more positive reminder.

Users are classified into three groups for the type of payment expected. A single user is entitled to use the software in the same fashion as a hardcover book, that is only one copy may be running on one machine at any one time.

My contact details are given at the bottom of this help page.

Group1 Single user full-time academic students, full-time academic staff, or unemployed.

Members of this group may continue to use this software without charge. If you would like to show your appreciation for this software in a stronger fashion then may I suggest one of the following:

- Send me a postcard from your part of the world to the address given below.
- If you feel very strongly about how much this program has changed your life and have a good book or two that you no longer read and want to pass onto a good home then please mail it/them to me at the address given below.
- If you want to part with hard earned cash then you may pay for ProgramBar as a Group 2 user.

Group 1 users who move up to Group 2 and have not registered under any of the bulleted points above are expected pay for this software as a Group 2 user.

Group 2 All other single users not in Group 1

Payment for ProgramBar is £10 (ten British pounds sterling) or \$15 (fifteen US dollars) whichever is more convenient. Cheque, postal order, or money order are acceptable. Cash is also acceptable but somewhat more risky to send through the post. If you would like a receipt then please indicate this when you send the money.

Group 3 Site license for multiple users

ProgramBar may be licenced for multiple users. A site is defined as a building or group of buildings with a unique *external* postal address. Please contact the author if you wish to negotiate on behalf of multiple sites.

The number of users is defined as either the number people that will use the software, or the number of machines (including networked workstations) on which the software may be used, whichever is the lower.

Usage is limited to those machines for which the company has purchased, hired, or leased. It does not include machines owned by employees of that company.

Number of users	Price per user	
	£ (British pounds sterling)	\$ (US dollars)
First user	10	15
2nd to 5th user	8	12
6th to 10th user	6	9
11th to 25th user	4	6
26th to 50th user	2	3
51st user and over	1	1.50
Unlimited users at one site	300	450

The price is capped at approximately 170 users to £300 or \$450.

An [order form](#) is provided for your convenience.

DISCLAIMER OF WARRANTY

THIS SOFTWARE IS PROVIDED FREE AND "AS IS" WITHOUT WARRANTY OF ANY KIND. THE AUTHOR FURTHER DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION REMAINS WITH YOU.

IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION EVEN IF THE AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

DISTRIBUTION

You may distribute this software freely on any electronic based medium including but not limited to: uploading to a FTP site, uploading to a BBS, downloadable from a WWW page. May also be distributed as part of a CD-ROM collection of shareware/freeware. Any distribution should not limit my rights (either explicitly or implicitly) to distribute this software using the same medium or any other. The software should be distributed as received in the original archive file.

The only charge a distributor may make shall be limited to the cost of the medium by which the software is delivered, and those incurred in handling/distribution. A small shipping and handling charge may be made if this software is distributed as part of a collection.

[Return to the contents page](#)

OTHER NOTES

ProgramBar is Copyright © 1995 Ian Jefferies. All rights reserved. All rights not explicitly licensed to the user are reserved to the developer.

If you write or publish an editorial about ProgramBar then the author would appreciate receiving a copy via either conventional or electronic mail.

The most recent version of ProgramBar may be obtained from the SimTel mirror archive at `ftp.coast.net` in the directory `SimTel\win3\desktop` or any of the mirror sites as `prgbr???.zip`, where `???` represents the version number. Versions 2.1 and earlier of ProgramBar were released under the filename system `prgbar??`.zip where `??` represents the version number.

Contact address:

Ian Jefferies,
24, Meredith Close,
Pinner,
Middx,
HA5 4RP,
England.

Tel: England (0)181-428-1466

Internet: `jeffers@teaching.physics.ox.ac.uk`

Windows, Windows95 and MS-DOS are registered trademarks of Microsoft Corporation.

All other product names may be copyrighted and registered or unregistered trademarks/trade names of their respective owners.

What makes a good book?

Well, thats always a matter of personal taste. I prefer to read sci-fi and sci-fact, fantasy and some cyberpunk, although horror does also make an appearance in my book collection. Technical manuals (computing in particular) are also of interest. If it's a book that you've found interesting, learnt from and/or enjoyed, have no further use for it, its written in English and want to pass it on to someone who might appreciate it then it's fair game.

Many international users might not have a book in English that that they can pass on. It is not my intent that you go and buy a book especially (although there is nothing to stop you), so feel free to send a postcard instead.

Order form for ProgramBar

This order form may be printed by selecting **File|Print topic** from the main menu. Typically I will not acknowledge receipt unless you include an e-mail address or specifically request it.

ProgramBar v2.0 registration form.

Mail to:

Ian Jefferies, Telephone: (0)181-428-1466
24 Meredith Close, E-mail: jeffers@teaching.physics.ox.ac.uk
Pinner, Middx,
HA5 4RP,
ENGLAND

Your details:

Name: _____
Company position: _____
Contact address: _____

E-mail address: _____
Day telephone number: _____

Purchasing details:

Single user license:	Quantity:	Price per unit	Total:
ProgramBar v2.0	_____	@ £10 each (ten British pounds)	_____
ProgramBar v2.0	_____	@ \$15 each (fifteen US dollars)	_____

Site license:	Payment in £ (British pounds)	Payment in \$ (US dollars)	Total:
2 to 5 users	1 user (£10) + £8 / extra user	1 user (\$15) + \$12 / extra user	_____
6 to 10 users	5 users (£42) + £6 / extra user	5 users (\$63) + \$9 / extra user	_____
11 to 25 users	10 users (£72) + £4 / extra user	10 users (\$108) + \$6 / extra user	_____
26 to 50 users	25 users (£132) + £2 / extra user	25 users (\$198) + \$3 / extra user	_____
51 users and over	50 users (£182) + £1 / extra user	50 users (\$273) + \$1.50 / extra user	_____
Unlimited users at one site	£300	\$450	_____

How did you hear about ProgramBar?

Further comments:

Troubleshooting problems in ProgramBar

If a problem occurs with ProgramBar you may want to consult this section of the help file so that you can find out how ProgramBar goes about performing certain tasks. This may save some hassle and help you solve the problem without having to consult the author.

*** Important ***

Some of these troubleshooting tips require looking at or modifying windows files. It is important that you are confident in making these modifications, and even more important that you make a backup of the files to be edited. This may save considerable premature graying of the hair should something go wrong. *You undertake suggested modifications at your own risk. If in doubt consult an expert in PC matters.*

A Windows INI file may have a line 'deleted' by placing a semicolon (;) as the first character in that line.

[Using ProgramBar](#)

[Return to the contents page](#)

[Reporting bugs in ProgramBar](#)

Problems during startup

ProgramBar looks at two INI files during the startup procedure, `progman.ini` and `control.ini`. All subsequent files looked at are listed somewhere in these files.

Looking at `progman.ini`

In the `[Settings]` section is an entry labelled `Order`. This is a long list of numbers that represent the order in which groups will be selected in Program Manager when using the keyboard interface. ProgramBar reads in this list of numbers and then expects there to be a corresponding entry in the `[Groups]` section. If there is no such entry then no further groups will be loaded.

Problem: Only a few groups are displayed by ProgramBar.

Solution: Check the `Order` entry in `progman.ini`. Look for a number without a corresponding entry in the `[Groups]` section. Check that all the group files mentioned in the `[Groups]` section actually exist in the Windows directory, the Windows system directory or on the path.

Problem: ProgramBar complains it cannot find a file during startup.

Solution: Run ProgramBar with the `/d` command line option:

i.e. `C:\WINAPPS\PROGBAR\PROGBAR.EXE /d`

replacing the path as appropriate.

This will generate a file called `progbar.log` in the directory containing the ProgramBar executable. This file contains information on the drivers and groups that ProgramBar attempts to load. Check this for any errors. For failed file loads check to see that the file actually exists on your hard-disk, either in the Windows directory, Windows system directory, or on the DOS path.

Problem: ProgramBar doesn't update the icons when they are changed in Program Manager. Groups and items are not updated on ProgramBar menus when remove/added in Program Manager.

Solution: ProgramBar only scans this information once - on startup. If your

Program Manager groups change then you will have to re-run ProgramBar in order to update the information. This may be fixed in a future release.

Looking at control.ini

ProgramBar looks at this file in order to provide the same functionality as Control Panel but from a popup menu. During startup ProgramBar builds up information about the items displayed by Control Panel, obtaining icons and informative title names. This information is found in the [drivers.desc] section. Some machines may not have this section, most likely because the machine does not have extra drivers installed.

The entries in the [drivers.desc] section give long titles for Control Panel to use for the drivers that might be loaded on the system. Each driver file is separated from its description by an equals sign. ProgramBar loads each driver (despite their extensions they are just dll's) and tries to initialize it as a Control Panel configurable item. If the initialization succeeds then the icon and title are obtained. The driver is then unloaded from memory.

Custom written Control Panel applications have a .CPL extension and do not appear in the [drivers.desc] section. ProgramBar searches for these files in the Windows directory and the Windows system directory only. The initialization procedure is identical to that of a driver.

Problem: ProgramBar complains it cannot find a file during startup.

Solution: Run ProgramBar with the /d command line option:

i.e. C:\WINAPPS\PROGBAR\PROGBAR.EXE /d

replacing the path as appropriate.

This will generate a file called PROGBAR.LOG in the directory containing the ProgramBar executable. This file contains information on the drivers and groups that ProgramBar attempts to load. Check this for any errors. For failed file loads check to see that the file actually exists on your hard-disk, either in the Windows directory, Windows system directory, or on the DOS path.

Problem: The filename in the error message is not in this log file.

Solution: Delete the file PROGBAR.DAT in the ProgramBar directory and try running ProgramBar again with the /d command line option.

If the error persists the one of the drivers listed in [drivers.desc] is attempting to load a file it expects to find on your hard-disk. Check to see if the file is on your hard-disk. If it is, but is not in the Windows directory, Windows system directory, or on the DOS path then try moving it into the Windows directory.

If you don't have a copy of the file then you most likely have an incomplete installation of a software package. Try to re-install to software package that used the driver. If you don't have the package then try removing the offending line in [drivers.desc]. You can find it by checking progbar.log to see which DLL or driver failed to load correctly.

Problem: This still doesn't solve the problem!!!

Solution: Try searching for an INI file that may be used by the driver/dll. There may be some path related information used by the driver/dll to find the missing dll.

Problem: I still can't solve it!!!&^£"\$&^£"!

Solution: Call in an *expert* PC user who knows and understands how Windows works. Failing that, panic!

[Using ProgramBar](#)

[Return to the contents page](#)

[Reporting bugs in ProgramBar](#)

Reporting bugs in ProgramBar

Every effort has been made to ensure that the program runs in a stable and usable manner. Unfortunately bugs do creep in occasionally, sometimes because another programmer has done something unusual in their code, sometimes because a user has an unusual configuration that ProgramBar cannot cope with, and occasionally because I've done something so stupid it defies belief.

The severity of the bug reported may result in an immediate release of another version of ProgramBar. It is in my best interest to protect the reputation of ProgramBar, a program that crashes or is unusable is worse than useless. If you have a look at the [version release information](#), you can see that this has already happened once...

In the unlikely event of ProgramBar crashing it is strongly recommended that unsaved data is committed to permanent storage before ProgramBar is run again. Due to the nature of ProgramBar's interaction with Windows, unexpected instabilities may occur in your system resulting in either lock out, further crashes of any application on the system, or an unexpected return to DOS.

[Return to the contents page](#)

[Shareware licensing and contacting the author](#)

Below are a some guidelines on reporting bugs that will help make my job of tracking them down and fixing them much easier. It will also save a long drawn out e-mail conversation.

- The version number of ProgramBar you are using.
- If the problem is caused by interaction with an application then the name of the application and its version number will be very helpful. If the program is shareware/freeware then a pointer on its availability on the Internet would be very useful in case I have to use some tools to study the problem. Please don't e-mail me the package unless I request it... my mail box might not be able to handle it.
- If the problem is due to a file not being found during startup, try running ProgramBar with the `/d` command line option. It will produce a list of all programs/dll's loaded during the startup sequence - and error codes if any - in a file called `progbar.log` in the ProgramBar directory. This file will provide me with more information and may help solve the problem.

In addition, startup problems may be due to an error in either `progman.ini` or `control.ini`. Sending me both of these files may also be useful.

- A full description of the problem and how to reproduce it. Possibly the best diagnostic tool available to me since I can't look over your shoulder or work at your machine. Consider the following two descriptions of the same problem:

"I get two buttons on ProgramBar. Can you help?"

"With ProgramBar running, launching Pegasus Mail v2.10 produces two buttons on the front panel. Only one window is on the desktop, but either button selects this window. After selecting Pegasus the

same button is always depressed regardless of which one was pressed. The extra button is not present if Pegasus is loaded before ProgramBar. Can you help?"

This was a real bug in ProgramBar (and had nothing to do with Pegasus Mail whatsoever). Which description would you prefer to receive in dealing with a bug? And you wonder why technical support staff get frustrated...

- Let me know if you would like to test the code I write to fix the problem. At a minimum it will require using PKUNZIP and UUDECODE, and you must be able to receive an e-mail of 150k or greater in size. You must also be able to extract the e-mail from your mail utility and get it to your PC. The file I send may well contain `new' features that I have added since I uploaded the latest release. In order to keep the traffic to a minimum, I'd prefer to deal with the first person who reports the bug and is prepared to test the code. Not wanting to test code is not a good reason for not reporting the bug! Someone who is more familiar with the ins and outs of Windows will also make life easier for both of us.

[Return to the contents page](#)

Things to do

This section details some of the plans that I have for ProgramBar for future releases. Many of you have contributed these ideas in your e-mail, ideas for new features are always welcome.

[Return to the contents page](#)

- Make ProgramBar run as the shell application. This has to be the number one request in e-mail that people have sent to me, and requires the most work.
- Adding a popup delay. Several people have said that they overshoot controls near the bottom of the screen (most notably scroll bars) and trigger ProgramBar instead. The time it takes for ProgramBar to hide itself again then becomes an understandable annoyance.
- A better task manager than the Task Manager Windows uses by default.
- A better Alt-TAB interface for those die hards...
- A hot-key to bring ProgramBar up from its hidden state.
- An optional document manager feature that groups documents by the executable that will be used to run them.
- User ordering of the items on the fast access menus.
- Alarms added to the clock.
- A full calendar in conjunction with the clock alarm feature.

