ProgramBar for Windows 3.1

Written by Ian Jefferies
(ProgramBar@eudoxus.demon.co.uk)

This release: 9 March, 1997
Version 2.30

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.

Whats new in v2.30

Overview

Introduction to ProgramBar

Installing ProgramBar

Using ProgramBar

Configuration of ProgramBar

Registration

Shareware agreement, order form, and statement of warranty

Miscellaneous

Obtaining the latest version

Contacting the author

Acknowledgements

Known problems with ProgramBar

Troubleshooting problems in ProgramBar

Reporting bugs in ProgramBar

Version release information

Wish list of things to do

Name
A descriptive name associated with the alarm. Used to build a descriptive title.

Enabled
Allows for individual alarms to be turned off quickly and easily.

Date
Displays the date that the alarm will next trigger. The format is that of the long format set in the International settings of Control Panel. This field cannot be edited, use the **Set date** button instead.

Set date
Changes the date on which the alarm will be triggered. A calendar dialog box is displayed to facilitate selection.

Hour
The hour on which the alarm will be triggered.

When in 24 hour format valid values lie in the range 0 to 23. When in 12 hour format valid values lie in the range 1 to 12.

Minute
The minute on which the alarm will be triggered.

Valid values lie in the range 0 to 59.

AM/PM
If you have a 24 hour clock then this control will not be present. Otherwise two values (default morning=AM, default evening=PM) indicate which half of the day in which the hour value is appropriate.

The choice of 12 or 24 hour representation is made when the dialog box is created.

Late alarm
Specifies the action to take when the alarm is triggered late. It is also possible to accumulate alarms over some period. Possible actions are:

Ignore	Ignore and delete
Trigger last	Trigger last and delete
Trigger all	Trigger all and delete

Repeat alarm
When selected the alarm will repeat with the specified

period.

Repeat frequency
Determines the next time that a repeating alarm will trigger. Possible repeat cycles are:

Every minute	Every 5 minutes	Every 10 minutes
Every 15 minutes	Every 20 minutes	Every half hour
Hourly	Daily	Weekly
Weekends	Weekdays	Fortnightly
Every 3 weeks	Every 4 weeks	Monthly
Quarterly	Yearly	

Enable chime
When selected a sound file will be played when the alarm is triggered.

Chime file
Displays the full path of the WAV file that will be played when the alarm is triggered. The file name may be edited directly.

Set chime file
Opens a search dialog that assists in selecting a WAV file to be played when the alarm is triggered.

Enable run command
When selected the command line specified will be executed when the alarm is triggered.

Executable
The executable and arguments that will be run when the alarm is triggered are placed here.

Documents may also be placed here and they will be run by their File manager associations.

The arguments may include environment variables. Enclose the environment variable with percent (%) symbols just as you would on a DOS command line. Use %% to obtain a single % symbol.

Browse executable
Opens a dialog that assists in searching for executables and documents.

Confirm execution
When selected a dialog box will be displayed requesting confirmation that the command should be run.

Run minimized
When selected the executable will be run as a minimized window. This will should not interrupt the current foreground task.

Useful for automated tasks that do not need the users attention.

Run maximized
When selected the executable will be run as a maximized window.

Enable message
When selected the message text will scroll across the title bar of the active window when the alarm is triggered.

Message text
Set the text that you want to be displayed when the alarm is triggered.

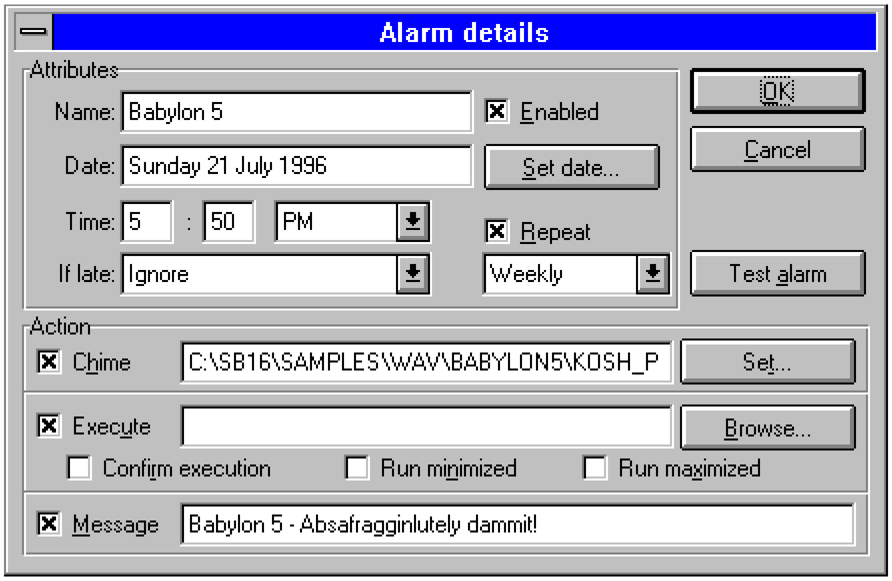
Test alarm
When pressed the alarm action will be demonstrated. This is particularly useful in ensuring that the command line in the execute section is accurate.

The message will always be shown in a window so that any scrolling messages will not be disturbed.

Front panel clock/date
When double clicked the front panel clock will open up a calendar window that covers 100 years.

Alarms

ProgramBar features   Using ProgramBar   Calendar



Name: Babylon 5
[✓ Enabled]
Date: Sunday 21 July 1996
Time: 5 : 50 PM
[✓ Repeat]
If late: Ignore
Weekly
[✓ Chime] C:\SB16\SAMPLES\WAV\BABYLON5\KOSH_P
[✓ Execute]
Confirm execution   Run minimized   Run maximized
[✓ Message] Babylon 5 - Absafragginlutely dammit!

This dialog sets the alarm parameters for triggering, and the action that will be taken when activated. When activated, a message may be scrolled across the title of the active window, sound a chime, run an application, or any combination of actions. Once triggered the alarm may then be reset to trigger again after some period or deleted. The alarm may also `trigger' while Windows is not running and this may be ignored or delayed until the next time Windows is running.

Alarms will only trigger on minute boundaries. The dialog box will verify that the alarm is valid before the dialog box is closed. It is not possible for an alarm to trigger while it is being edited. The alarm may however be tested before the dialog is closed.

The time format will be either 12 or 24 hour. This will be determined by the International settings used by Control Panel when the dialog box is created.

New alarms may be created from the calendar or the configuration panel.

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.

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 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.

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.

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.

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

+Fast access
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 to File Manager will not be added.

+Desktop
Adds the selected file(s) to the desktop as an iconic link.   Enabled only when the selection has been made and the desktop is live.   Files that do not have associations known to 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.

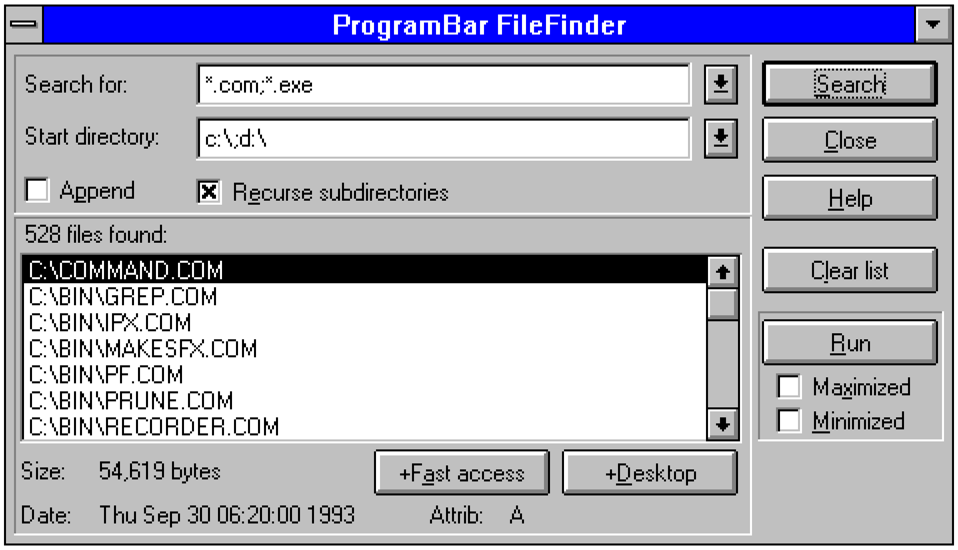
Run minimized
When checked, programs that are run by selecting the `Run' button will run as icons.   This button does not affect how items are added to the fast access menus using the `Add' button.

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.

The *Start* menu: Find file...

ProgramBar features   Using ProgramBar



Search for: *.com;*.exe
Start directory: c:\;d:\
☐ Append ☒ Recurse subdirectories
528 files found:

```
C:\COMMAND.COM
C:\BIN\GREP.COM
C:\BIN\IPX.COM
C:\BIN\MAKESFX.COM
C:\BIN\PF.COM
C:\BIN\PRUNE.COM
C:\BIN\RECORDER.COM
```

☐ Maximized
☐ Minimized
Size: 54,619 bytes
Date: Thu Sep 30 06:20:00 1993 Attrib: A

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, executed, or added to the live desktop.

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' check box 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.

Three actions may be performed on the selected files. First, then may be added to the fast access menus for applications, documents or help files by clicking on the `+Fast Access' button. Secondly, they may be added to the documents on the live desktop by using the '+Desktop' button. Finally 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, added to the fast access menus or the desktop. 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 full screen. 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 :)

Year and month
Indicates the year and month for which information is displayed in the list box

Day of week
Marks the column for which each day in the list box belongs.

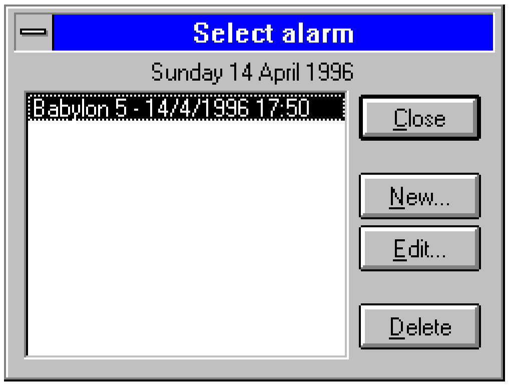
Dates
The currently selected date is highlighted in black.   A date that has an alarm associated with it will display red text.

Double-click on a date to add or edit an existing alarm.

Close
Closes the calendar dialog box.

Today
Returns the selected date in the calendar dialog to today.

Edit alarm...
Displays a list of the alarms associated with the selected date.



Sunday 14 April 1996

Babylon 5 - 14/4/1996 17:50

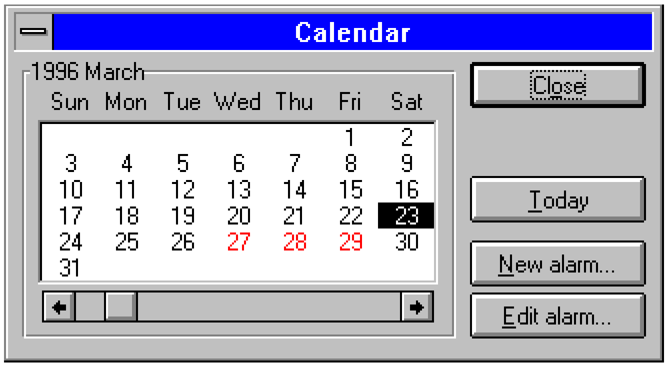
New alarm...
Adds a new alarm at the selected date, and presents a dialog box for configuring the alarm properties.

Calendar slider
Use this slider to select the month and year for which details are displayed by the calendar.

Pick up and drag the slider to choose the required month/year. Use the arrow buttons to move forward or backward one month. Click on any other part of the scrollbar to advance or reduce the date by one year.

ProgramBar Calendar

ProgramBar features   Using ProgramBar   Alarms



Select a control on the dialog box for more information.

The calendar assists in finding the correct date for a particular day.   It has a range of 100 years, beginning with January 1st 1970.

The calendar may also be used to add/edit alarms.   By double-clicking on a particular date an alarm may be added.   If an alarm is already set for that day (indicated by a red character) then a list allows one to be chosen/added or deleted.

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.

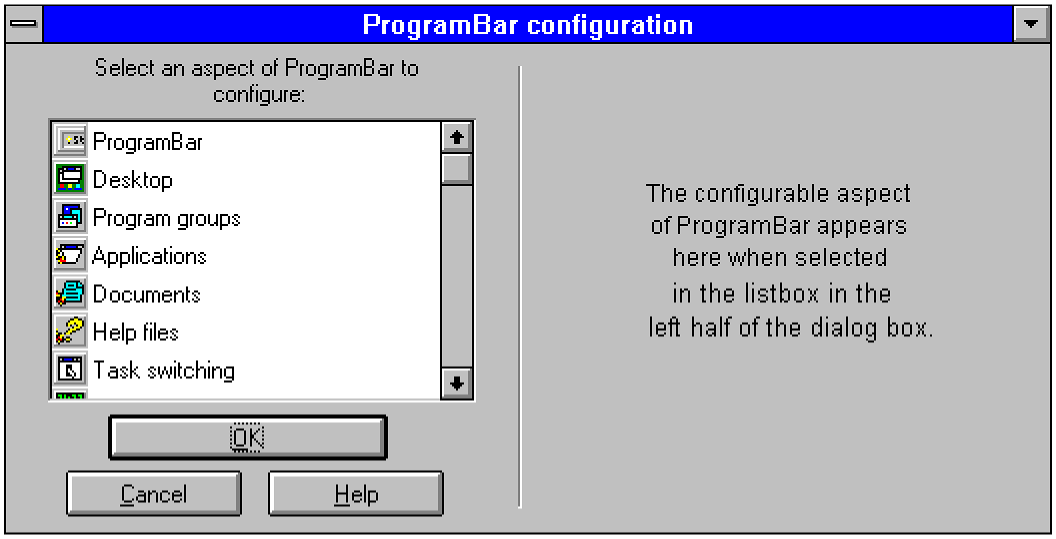
Configurable aspects of ProgramBar
Select an item in this list box in order to adjust its properties. The properties will appear on the right hand side of the configuration dialog.

Display list of configurable properties

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

Configuration of ProgramBar

Introduction Using ProgramBar Configurable items


ProgramBar configuration
Select an aspect of ProgramBar to configure:
ProgramBar
Desktop
Program groups
Applications
Documents
Help files
Task switching
OK
Cancel
Help
The configurable aspect of ProgramBar appears here when selected in the listbox in the left half of the dialog box.

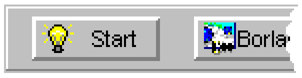
The configuration menu may be accessed by selecting Configuration|ProgramBar from the Start menu. A dialog box appears, split into two parts. The left hand side present a list of the aspects of ProgramBar that can be configured by the user. Selecting one of the items on the list will display an appropriate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

Select a section of ProgramBar that may be configured.

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 version 2.0 onward all changes to `progbar.ini` can now be made through ProgamBar 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!

The Start button

ProgramBar features   Using ProgramBar



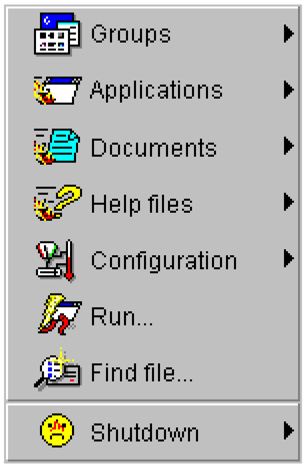
This button on the left hand side of the bar gives access to much of the functionality of ProgramBar. Through it you may access the items in your Program Manager groups and load them.   You can also configure any part of Windows that may be changed through the Control Panel application.

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.

The *Start* menu

ProgramBar features   Using ProgramBar

The menu accesses the extra functionality of ProgramBar.   Click on the menu item you want to know more about.   You may also select the item by using TAB and Shift-TAB.



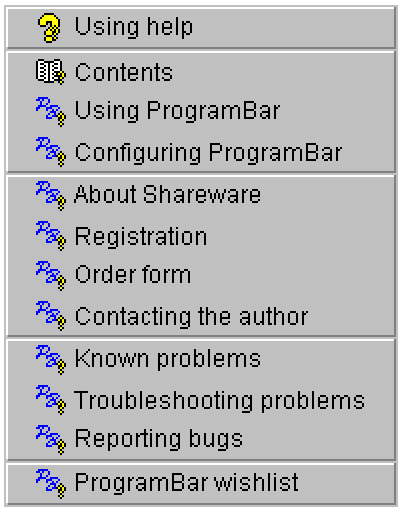
ProgramBar logo

ProgramBar features Using ProgramBar

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

ProgramBar logo - context menu

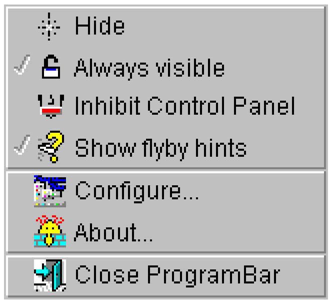
ProgramBar features Using ProgramBar



This menu is obtained by right clicking on the logo on the front panel. It gives direct access to items within the help file system.

System menu

ProgramBar features Using ProgramBar



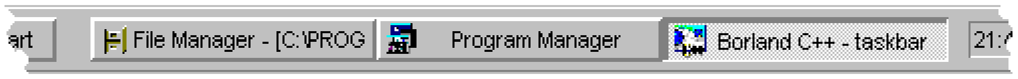
A single click with the left mouse button on a part of the ProgramBar panel other than a button will display the system menu. From this menu you may close ProgramBar, or set whether it is always visible at the bottom of the screen.

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!)

Currently running applications

ProgramBar features   Using ProgramBar



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.

Right clicking on an application button shows a context sensitive menu detailing other actions that ProgramBar can do with that window.

Activate
The application will be brought to the foreground, restored from iconic state if necessary.

Move to virtual screen
The application will be moved to the specified virtual screen. It is not possible to move an application to the same virtual screen it is already in.

If the application is iconic, it will be restored.

State of application window
Change the window state to one of Maximized, Restored or Minimized. The current state of the window is disabled.

Performing one if these actions on the application window will give it focus.

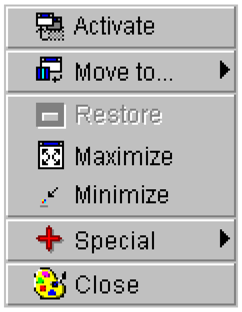
Special operations
A submenu that lists specialized operations on the application window.

At present the only operation is adding the application to the task exclusion list.

Close application
Closes the application in an identical fashion as the close window menu.

Currently running applications - context menu

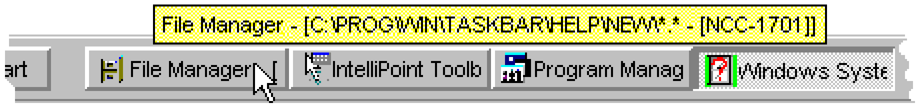
ProgramBar features Using ProgramBar



This menu is displayed when the right mouse button is clicked on the application button on the front panel of ProgramBar. It allows you to modify the state of the window in conjunction with its appearance on the screen and how it interacts with ProgramBar.

More information on applications

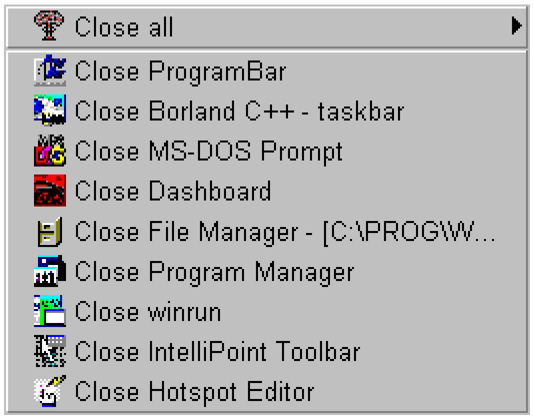
ProgramBar features   Using ProgramBar



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.

Close window menu

ProgramBar features   Using ProgramBar



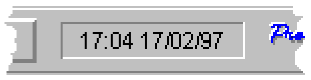
A click of the right mouse button on a part of the ProgramBar panel other than a button will display a menu that allows you to close any visible window running on the system.   Keep the right mouse button pressed down as you drag the mouse pointer to select the required menu option.   From this menu you can also close all applications that are either minimized, maximized or running in a window smaller that the full screen.

If the window is the main window of the application then the entire application will be shutdown.   In some cases (e.g. Netscape) several main windows may be open and closing one window will not affect the remaining windows.   In this case the application will be shutdown when the last window is closed.

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!**

Clock and Date

ProgramBar features Using ProgramBar



The clock may be displayed in the front panel font or in a digital format. When the front panel font is displayed the date may also be shown in the short format defined in the Control Panel International settings. It is not possible for the date to be displayed with the digital clock.

When the flyby hints are active, hovering the mouse over the clock will give the current date in the long format defined in the Control Panel International settings.

Double clicking on the clock will display a 100 year calendar.

Edit alarm
Presents the list of alarms so that one may be selected for editing.

New alarm
Create a new alarm.

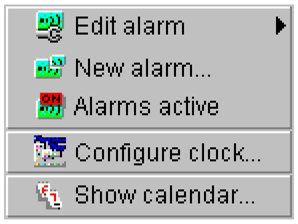
Alarms active
Enables or disables all of the alarms. This is useful if you do not want to be disturbed by an alarm window popping up.

Configure alarms
Opens the configuration dialog at the clock and alarms configuration page.

Calendar
Opens a 100 year calendar.

Clock - context menu

ProgramBar features   Using ProgramBar   Calendar



The context sensitive menu may be activated by right clicking on the clock on the front panel.

The menu allows existing alarms to be edited, each alarm will be identified by its title and the next time it will trigger. A new alarm may also be created.

The alarm system may be disabled temporarily. This allows you to work without being disturbed by an opening dialog box or a change in an applications title.

In order to assist with the configuration of the clock, the configuration dialog box may also be opened. It will be opened at the page that edits the clock and alarm settings.

A calendar can also be shown - this is a complementary method for opening the calendar and is equivalent to double-clicking on the front panel clock.

Virtual Screens

Virtual screens effectively increase the area of the screen available for the user to place applications on. Between 2 and 7 screens may be used, each screen accessible via the front panel or use of the (user definable) hotkey.

Probably the easiest way of thinking of the virtual desktop is as a faster Alt-TAB.   By placing all your applications on separate screens you save having to hunt for them through the Alt-TAB interface. Windows may be moved between virtual screens via the right click menus on the front panel.

Each of the virtual screens behaves just like the single desktop area that is normally available to you. Applications that have been reduced to iconic form are present on all virtual screens.

It is also possible to force some applications to remain fixed at the same position on all the screens. These applications are "sticky" windows.

Live desktop

ProgramBar features   Using ProgramBar

ProgramBar provides significant enhancements to the standard Windows 3.1 desktop. These attributes are enabled and set through the configuration dialog. The functionality is intended to be as unobtrusive as possible while making Windows a better environment to work in.

- Iconic document links may added to the desktop, allowing quick access to frequently used documents.
- Application and document icons may be moved to any edge of the screen rather than leaving them on the bottom edge.
- The icons may be automatically arranged, preventing gaps being left when icons are restored or closed.
- The desktop now has a system menu that gives access to desktop related functionality through a single right click.

Title
The long title associated with the file.

File
The filename that the link is associated with. It may be changed by using the **Browse...** button.

Title displays
This combination box sets the format of the title string of the document link. There are three possible formats:

Title
Filename and path
Filename only

Allow copy
When enabled the icon may be copied freely. When disabled the icon cannot be copied.

Desktop document links

ProgramBar features		Using ProgramBar

Files may be dragged from File Manager onto the desktop. The files that have associations will have an document link created. Alternatively, the file finder may be used to locate and add file to the desktop. The desktop system menu can also be used to add files to the desktop.

The document link behaves like an iconic application window, it can be picked up and moved, closed and activated. A small picture of a document appears in the bottom left hand corner to indicate that the icon is a link rather than a minimized application.

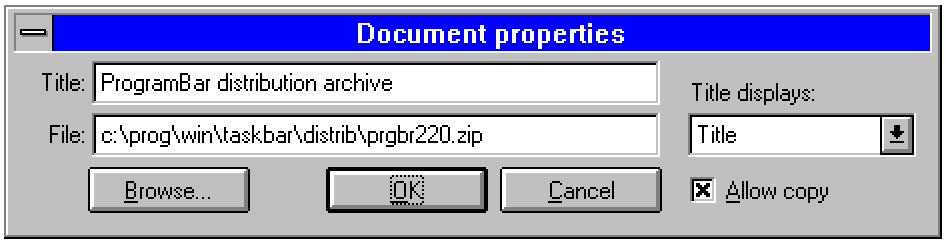
Document links represent a resource friendly way of having documents available for editing without loading a large program. They complement rather than replace the fast access menus provided in previous versions of ProgramBar. Since each document may have a title associated with it, a better description than the filename may be given. This is similar to the long filenames employed by Windows 95.

Editing the document
Double click on the icon to run the associated application and load the document for editing.

Copying the document link
With the CTRL key pressed down it is possible to create a copy of the link by clicking on the icon and dragging it to its new position on the desktop. Alternatively open the system menu and select copy on the menu. The link must be enabled for copying.

Activating the system menu
A single click on the icon produces a system menu from which the document may be edited, deleted, moved, copied or have its properties adjusted.

Editing the properties of the link
Right click on the icon to quickly access the properties dialog.


Document properties

Title: ProgramBar distribution archive

File: c:\prog\win\taskbar\distrib\prgbr220.zip

Title displays: Title

[✓ Allow copy]

Desktop icon positioning

ProgramBar features Using ProgramBar

The application and document link icons may be arranged onto the same or separate edges. Icons are no longer limited to the bottom edge of the screen and may be placed according to the users preference.

When program and document link icons are placed on the same edge they will mix freely. If they are placed on separate edges then the application icons will always take precedence in their positioning.

The icon arrangement for applications and documents may be changed in the configuration dialog.

To obtain a Windows 95 appearance move both the application and document link icons to the left edge.

Desktop automatic icon arrangement

ProgramBar features Using ProgramBar

When a minimized application window is restored or closed this normally leaves a gap amongst the iconic windows. The desktop may be set to automatically arrange the iconic windows so that your desktop is always neat without any effort on your part.

The automatic icon arrangement may be turned on or off in the configuration dialog.

The desktop system menu may be used to arrange the iconic windows.

* * * Important * * *

It is possible for another application to re-arrange the icons without ProgramBar noticing. This occurs when the `ArrangeIconicWindows` function is called for the desktop window. The default Windows task list (activate with CTRL-Esc) behaves in this way. Always use the desktop system menu to arrange icons. ProgramBar will recover when a desktop icon is selected, but the order of the icons may be scrambled.

Desktop system menu

ProgramBar features Using ProgramBar

A right click on the desktop will display a popup menu from which desktop properties may be accessed.

The application and document link icons may be arranged. It is also possible to turn the automatic icon arrangement on or off. Document links may be added to the desktop, and the configuration dialog displayed or brought to the fore.

The screen saver may be triggered or temporarily disabled. When ProgramBar is shutdown the screen saver will be returned to the state of activation it was in before ProgramBar was run.

The Windows task list may also be activated.

The *Start* menu: Groups

ProgramBar features Using ProgramBar

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.

The *Start* menu: Applications

ProgramBar features   Using ProgramBar

When selected a popup menu appears with a list of applications set up 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 removed by use of the configuration dialog box.

The *Start* menu: Documents

When selected a popup menu appears with a list of documents that have been set up for quick access. It is suggested that you place all the frequently accessed documents you use here so that they can 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 its 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 removed by use of the configuration dialog box.

The *Start* menu: Help files

When selected a popup menu appears with a list of help files that have been set up for quick access. It is suggested that you place all the frequently accessed help files here so that they can 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 removed by use of the configuration dialog box.

The *Start* menu: Configuration

ProgramBar features Using ProgramBar

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.

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 full screen.

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.

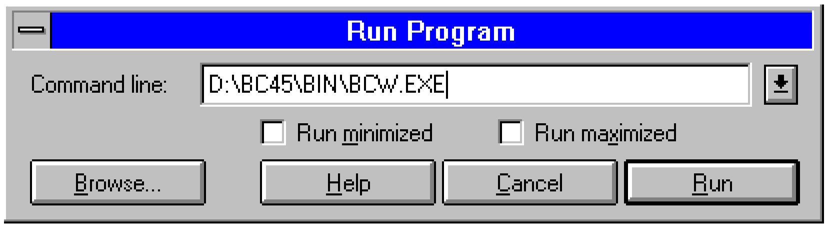
Cancel
Cancels the dialog box.   No program will be run and the contents of the Execute combo box will not be added to the history list.

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.

The *Start* menu: Run...

ProgramBar features   Using ProgramBar


Run Program

Command line: D:\BC45\BIN\BCW.EXE

☐ Run minimized   ☐ Run maximized

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 full screen 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.

The arguments passed to an executable may be environment variables. Enclose the environment variable with percent (%) symbols just as you would on a DOS command line. Use %% to obtain a single % symbol.

Click on the Run 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.

The *Start* menu: Shutdown...

ProgramBar features Using ProgramBar

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.

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. If ProgramBar is running as the shell application then Windows will be shutdown and returned to the DOS prompt. You will be prompted to save changed documents.

Whats new in ProgramBar v2.30

To help you find out what is new to this version, this page has been added to the help file. It gives a very brief description of the new features, changes to existing features, and a pointer on where to find out more.

Virtual screens
Designed to help you lay out your applications over a larger area. Each virtual screen can be assigned a hot-key for quick access through the keyboard. The virtual screens can also be accessed as part of the usual front panel interface. It is possible to hide the layout of the virtual screens on the front panel while still using the hot-keys for access.

Context menus for front panel buttons
Many of the buttons on the front panel now have a context menu that can be accessed by a right click of the mouse on the appropriate front panel area.

Three such menus are currently implemented: they complement the task buttons that allow you to switch between applications, the functionality of the clock and alarms, and give faster access into components of the help system.

New alarm features
Several new features have been added to extend the functionality of alarms.

A larger range of time intervals can now be set: preset periods of less that one hour are now available (1 min, 5 min, 10 min, 15 min, 20 min, 30 min), as well as several longer periods (3 weeks, 4 weeks).

When a program is run by an alarm it's execution can now be confirmed, it can also be run minimized and maximized.

Alarms may be tested before the alarm edit dialog is closed.

Reduced load time
ProgramBar now loads even more quickly than previous versions. Data from the Program Manager groups is more heavily cached. It is necessary to rebuild the cache each time the screen resolution is changed. It can also be manually flushed. This performance change should be most noticable for 386 and 486 systems.

Windows 95 visual appearance
This option makes the front panel look more like the Win95 taskbar. Front panel buttons take on a square edged appearance, task buttons are slightly separated from each other, and menus use a similar blue highlight to indicate the selected menu item.

This option is set in the configuration dialog box under *ProgramBar - General*.

Reversing of program titles
For those programs that place a filename in their window title in the form ***Application name - filename***, ProgramBar can now reverse this to show the filename first. Useful for when you have the same

application open several times.

This option is set in the configuration dialog box under *Task switching - Preferences*.

Click-drag operation of the Start button
Many users have requested that the Start button allow a "click, move, then select menu item" as well as the "click and drag" model currently implemented. This has now been implemented (within the restrictions of the Windows 3.1 menu system), and can be set in the configuration dialog box under *ProgramBar - General*.

Other changes
A more complete list - including bugs fixed - is available by looking at the version release information.

Introduction to ProgramBar

Installing ProgramBar   Using ProgramBar   Configuration

Windows was designed so that you might run several programs together.   While doing this you quickly find that you run out of screen real estate.   Reading a help file while writing and testing an Excel based macro for example, or more.   Switching between applications now means 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 are implemented in this release:

- Fast access menus for your most frequently used programs, documents and help files.
- 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.
- A live desktop allowing frequently used documents to appear on the desktop as icon links.   Documents may be dragged from File Manager and added to the desktop.   A desktop system menu is available by right clicking on the desktop.
- Access to all of the Control Panel configuration applets.
- Each running application represented on the front panel as a button.   A push of the button makes that application active. Iconic applications may be hidden so as to neaten the desktop.
- Right clicking on the ProgramBar panel displays a list of active applications that can be closed.
- Shutdown Windows, either returning to DOS, restarting Windows again, or rebooting your machine.
- Fast search for files on your hard disk.   They may be launched or added to the fast access menus or live desktop.
- Run a program specifying the command arguments, or a data file with an association known to File Manager.
- The current time and date displayed on the front panel.   A calendar window helps you plan and set alarms.
- Full configuration of ProgramBar via a popup dialog box.
- Ability to run as a limited shell.   No support for DDE.

- Virtual desktop so that you can increase the effective size of your screen and switch between applications quickly and easily.

- Right click on the front panel controls for context sensitive menus that add functionality.

Other features that will help you make better use of Windows:

- Colourful icons help navigation through the menus.

- Flyby help hints when you let the mouse hover over part of the ProgramBar interface.

- Quick access to this help file (click on the ProgramBar logo on the right hand side of the bar.)

- 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.

- When permanently visible maximized windows may be resized to accommodate the reduced desktop area.

- Applications may be excluded from the front panel task switching buttons to reduce clogging up of the panel.

- Task switching buttons on the front panel may be stacked onto more than one row.

- ProgramBar may be triggered from any of the four edges of the screen. A convenient marker can be displayed on the last triggered edge, or all available triggering edges.

- 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.

- Better support for people who use cursor screen wrap: the size of the region that can be used to make ProgramBar popup may be adjusted. The time delay before ProgramBar hides again may be increase. Also a time delay before ProgramBar is displayed may be set, avoiding the `overshoot' problem associated with controls near the edge of the screen.

- 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.

- 3D effect on all dialog boxes though the automatic use of CTL3DV2.DLL if present. Tabbed dialog boxes are adjusted to look more like their Windows 95 counterparts when used in conjunction with this DLL.

- Better crash protection. ProgramBar interferes with or supersedes much of the functionality of Windows in order to implement its features. Should another application `fall over' or crash ProgramBar should not be affected. Neither should ProgramBar cause another application to crash. In the unlikely event that ProgramBar crashes it removes all of its hooks and cleans up after itself, thus minimizing the chance of another program crashing.

- Hidden out of the way when you don't need it. May also be permanently visible on your desktop.

- The Control Panel may be inhibited from running in order that a potential clash between ProgramBar and Control Panel is avoided. The clash arises because the same dialog box may be opened by both applications and different values set. This is a precautionary measure only.

Installing ProgramBar

Using ProgramBar   Configuration

There are four 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.
CONFIG.HLP	Help file for the configuration dialog box.

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.

Install over a previous version
Installing ProgramBar as the shell

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`.

Details on alarms will be stored in `PROGBAR.ALM`, and information on the live desktop in `DESKTOP.DT`. The user may create additional desktop related files with the extension `.DTL`.

A cache of icons will be stored in the file `ICON.DAT`, this will reduce the load time for ProgramBar considerably by storing the small version of each Program Manager icon.

ProgramBar assumes that both the Program Manager (`progman.exe`) and WinHelp (`winhelp.exe`) executables are available. Some default icons are taken from these files for display purposes.

Installing over a previous version

Follow these instructions for a hassle free upgrade over a previous version:

1. Shutdown the version of ProgramBar currently running on your machine.
2. Make a backup copy of all the files in the ProgramBar directory.
3. Make a backup copy of `progbar.ini`, found in the Windows directory.
4. Copy all the new files into your ProgramBar directory.

To undo the new installation, delete all the files in the ProgramBar directory and `progbar.ini` in the Windows directory. Copy back your previous version.

Installing ProgramBar as the shell

Shutdown Windows and manually edit the `system.ini` file in your Windows directory. In the section labelled as `[Boot]` find the line that begins `shell=` and comment it out (insert a semicolon as the first character on the line). Now insert a new line `shell=<path to ProgramBar>\progbar.exe`, where <path to ProgramBar> is the directory you installed ProgramBar into.

To remove ProgramBar as the shell: comment out the inserted line and remove the semicolon from the original `shell=` line by manually editing the `system.ini` file while Windows is not running.

Using ProgramBar

ProgramBar runs in one of two modes, permanently visible or hidden when not required. When permanently visible the ProgramBar will not 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.

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 controls present. 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.

The front panel of ProgramBar

- **Start** button
    - Menu
- ProgramBar logo
    - Context menu
- System menu
- Task switching
    - Information on applications
    - Context menu
- Close window menu
- Clock
    - Calendar
    - Alarms
    - Context menu

Other components of ProgramBar

- Configuration
- Finding files
- Run command
- Desktop
    - document links
    - icon positioning
    - automatic arrangement
    - system menu
- Virtual screens

Activate the Program Manager application
You just activated the application with the title `Program Manager'. 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.

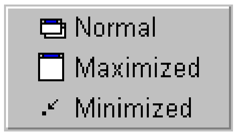
The most useful aspect of this at present is for the task switching buttons. Their flyby hint gives the full title of the window that will be selected. Normal operation of these buttons may truncate the title and make identification of similar programs difficult.

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.

Configure ProgramBar
Opens the configuration dialog, or brings it to the foreground.

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.

Known and potential problems with ProgramBar

Using ProgramBar   Troubleshooting problems   Reporting bugs

This section of the help file describes some problems that ProgramBar is known to have, or may potentially be involved in. In addition, some discussion is given to aspects of ProgramBar that may not be of general satisfaction.

Identifying applications for task switching

Closing down DOS sessions

Finding the icon used by an application

Borland Dashboard v2.0 compatibility

Virtual desktops

Providing a keyboard interface

Subclassing windows

Using ProgramBar as the Windows shell

Not all Program Manager groups displayed
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.

Cannot find a file during startup
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.

Program Manager groups don't match menus
Open the configuration dialog and select Program Groups. Click on the button that forces a rescan of the Program Manager groups.

No relevant details present in the log file
Delete the quick load file PROGBAR.DAT and try again.

Cannot find a file during startup
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.

When the problem file is located, delete the appropriate line from `control.ini`.

If there are no apparent errors then try deleting the file `PROGBAR.DAT` and running ProgramBar again. The failed filename might not match the file that failed to load.

Share violation when loading a desktop layout
When a desktop layout is loaded, ProgramBar checks for the existence of the document file. This will fail if the document is already loaded into an application that has used SHARE to prevent other applications from using reading the file. Microsoft Word 2.0c exhibits this behaviour.

The document link will be lost.

Invalid drive or file not found
The network drive that the document file is stored on has not been connected to, the network drive letter has changed between sessions, or the file has been deleted.

The document link will be lost.

Program fails to run from a menu or dialog
Run ProgramBar with the `/d` command line option and try to reproduce the error.

The filename, working directory and executable that are used in attempting to run the file or program will be listed. Use this to try and diagnose the problem.

Network users may experience failure due to not being logged onto the correct drive.

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.

ProgramBar can write a log file to assist in debugging many problems. This log file contains details on what files are loaded during startup, and how programs are run from the Group and fast access menus. To create the log file, run ProgramBar with the `/d` command line option.

* * * 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 greying 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.

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.

Problems related to `progman.ini`

Only a few groups are displayed by ProgramBar

ProgramBar complains it cannot find a file during startup

ProgramBar doesn't update the icons when they are changed in Program Manager

Groups and items are not updated on ProgramBar menus when removed/added in Program Manager

Problems related to `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 `[MMCPL]` and `[drivers.desc]` sections, the latter may not be present on some machines.

The entries in `[MMCPL]` and `[drivers.desc]` section give long titles for Control Panel to use for the drivers that might be loaded on the system. ProgramBar checks to see if the driver is loaded in memory. If it is then it requests information on whether it can be configured.

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.

Files are reported as missing during startup

Miscellaneous startup problems

ProgamBar reports a SHARE violation for a document link

ProgramBar reports a file not found or invalid drive

Problems when ProgramBar is running

Executing a program fails

ProgamBar reports a SHARE violation for a document link

ProgramBar reports a file not found or invalid drive

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.

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 250k 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.

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 2.30, released 9-Mar-97

Version 2.20, released 24-Apr-96

Version 2.11, released 1-Dec-1995

Version 2.1, released 21-Nov-1995

Version 2.0, released 18-Nov-1995

Version 1.2, released 23-Oct-1995

Version 1.1, released 22-Oct-1995

Version 1.0, released 9-Oct-1995

Things to do

Contents

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.

- 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.
- 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. Adding a keyboard interface.
- 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.
- Gizmos: Small buttons on the front panel that give access to a single tool e.g. CD player, DLL unloader, tearaway resource monitor etc.
- Full SDK for externally developed add-ons. Full access via DDE to the tools used within ProgramBar e.g. adding items to any of the menus, new tools on the front panel etc.
- System resource monitoring and low resource alarms.

Shareware, some legalese, and other notes

What is shareware?
Registration details
Order form
Contacting the author
Obtaining the most recent version

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.

OTHER NOTES

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

Verson 2.30, released 9-Mar-97

Version information

- Items on the Fast access menus will now appear as disabled if ProgramBar cannot find the executable/document or help file on startup.
- When used as the shell, ProgramBar now checks progman.ini for the group title that represents the StartUp group. If no group is specified then the name is obtained from the Program Manager executable. This varies across International versions of Windows.
- Stack fault in some Control Panel applications caused by interaction with CTL3DV2.DLL now fixed. ProgramBar no longer uses the AutoSubclass feature of this DLL. The side effect of this is that these appletss may not have a 3-D appearance.
- Control Panel applets that used the old interface (CPL_INQUIRE) are now recognised correctly (I finally found an example to test!).
- Better support for applications that use the desktop window to provide their functionality. All mouse messages are now passed to the desktop.
- Dialogs created by ProgramBar will no longer `underlap' the front panel when it is permanently visible.
- Crash when trying to parse the environment variable string. Triggered under specific circumstances. Fixed.
- Alarms were not updated and saved if another application was used to close down Windows. Fixed.
- Alarm message in dialog box state not always saved across sessions. Fixed.
- Occasional lockup of the ProgramBar front panel and desktop, caused by interaction with DOS windows or the use of Alt-TAB. Fixed.
- Maximizing a DOS window moved it to the top left of the screen. Fixed.
- Applications that were both minimized and hidden failed to appear on the close application menu (right click on front panel). Fixed
- The absence of a working directory in a Program Manager group caused an incorrect path to be constructed under certain circumstances. Fixed.
- When creating a copy of a document link the copy always appears in the bottom left corner (or nearest possible position), regardless of which edge the document links should appear on. Fixed.
- Document links created by copying an existing link were incorrectly placed on the desktop. Fixed.
- Desktop system menu now has the option of pressing CTRL,ALT and/or SHIFT in addition to a mouse button in order to activate. Added for

compatibility with other desktop menu software products.

- No error messages when a document link became invalid, or invalid documents were added to the desktop via the desktop menu. Fixed.
- Desktop icons were not moved if ProgramBar was permanently visible but autoarrange disabled. Occured during startup or change of ProgramBar to/from permanently visible. Fixed.
- A cache of the small icons used on the Group menu is now created. This greatly reduces the load time of ProgramBar.
- Startup splash screen now shows the current stage of ProgramBar's initialisation.
- When used as the shell ProgramBar now runs the command line passed to the windows loader `win.com`.
- Alarms may now be tested before being confirmed. This is particularly useful when programs with command line options are being run.
- When an alarm triggers an executable this may now be confirmed. In addition the window associated with the application may be run minimzed or maximized instead of its default size.
- Access to the Windows Setup program via the Start|Configuration menu.
- Extra range of alarm repeat periods added. 1, 5,10, 15, 20 and 30 minute repeat periods. Also added 3 week and 4 weeks repeat periods.
- The application window title displayed on the front panel may now be reversed so that "X - Y" may now be displayed as "Y - X" instead. This is useful for applications that display a filename/document in their window title.
- When an iconic application was activated by using the system menu to restore or maximize the window, ProgramBar failed to acknowledge this. Fixed.
- The front panel task switch buttons now have a popup menu associated with them that allow individual window manipulation - maximization, minimization, activation, closure and others. The menu is accessed via a right click of the mouse button on the button.
- The Hide item on the system menu failed to operate correctly when the front panel was permanently visible. Only maximized windows that had been reduced to the available desktop were affected. Fixed.
- When the front panel clock was not displayed ProgramBar occasionally locked up during startup. Fixed.
- The front panel clock now has a menu accessed through the right mouse button. All of the alarms may be edited, a new alarm added or the calendar shown.
- Added a virtual desktop. The effective screen area of Windows may now

be increased by up to seven times. A hotkey may be used to activate a given virtual screen. Applications may be defined as `sticky', all ProgramBar windows and iconic windows are treated as such by default.

- When closing Windows the live desktop documents were not saved if they had been modified by using drag/drop. Fixed.

- The load= and run= command lines of WIN.INI were not run correctly when ProgramBar was used as the shell. Fixed.

- Win95 visual appearance added.

- Improved full screen DOS box detection: prevents ProgramBar windows (which should not be switched to) from appearing on the DOS task switch list.

- Selection of task switching stack sizes less than 8 failed to register correctly on the configuration dialog. Fixed.

Version 2.20, released 24-Apr-96

Version information

- If a file specified in control.ini did not exist then ProgramBar crashed during startup. Fixed.
- ProgramBar refused to accept an existing file when editing the attributes of a file on the fast menu that was not on the current drive. Fixed.
- ProgramBar crashed under certain circumstances when trying to obtain the date time stamp of a non-existent file during startup. Fixed.
- Problems experienced by several users when loading ProgramBar: one or more DLL's complained of missing files or refused to be loaded a second time (warning messages only). Fixed.
- Iconized apps locked down the active focus button, preventing switching to that task from the front panel. Fixed, iconic apps no longer lock the active focus button.
- With ProgramBar permanently visible maximized windows may be resized to fill the available desktop. Multiple Document Interface (MDI) iconic windows may be forced to rearrange so that icon titles do not disappear off the bottom of the window area.
- Iconized application windows may be hidden from view only if they appear on the button bar. This neatens up the desktop. Iconic applications that do not appear on the task switch bar are not hidden, they may use their icon to present the user with information.
- Front panel task switch buttons incorrectly sized under certain circumstances. Fixed.
- While entering the password for a screen saver it was possible to activate ProgramBar. Fixed.
- When deleting the details of an application excluded from the button bar, sometimes the wrong item was deleted. Fixed.
- Dragging of a file onto ProgramBar caused update problems when the configuration dialog box was open and an Application/Document/Help list box required updating. Only triggered when the dragged file was inserted immediately before the current selection. Fixed.
- The fast access applications, documents and help files now use the directory associated with the executable/data file when no working directory is specified.
- ProgramBar load time reduced by caching .GRP files in memory. Should be most noticeable for people who have lots of large groups.
- Compatibility with Wayfarer shell introduced by popular request. The Wayfarer icon will now appear on the front panel as an application and it

can be activated. One side effect of this is that all applications that only appear in iconic form will also appear on the task switching front panel (they can be excluded). If selected from the front panel these iconic applications may show their system menu.

- Dialog boxes converted to use of a non-bold font - closer appearance to the Win95 dialog box style and more space in the text edit fields.

- The last edge used to trigger ProgramBar, or all the edges that may be used to trigger ProgramBar may be indicated on the desktop, in the same fashion as the Win95 task bar.

- By popular request the main menu may be configured to appear when the Start button is clicked instead of using the click-drag approach previously employed.

- Popup delay now implemented so that the cursor can now `bounce' off the edge of the desktop when using corner controls without triggering ProgramBar.

- The live desktop has its own popup menu (right click on the desktop) which includes the ability to arrange iconic windows, enable/disable the screen saver, run the screen saver, and bring up the configuration dialog box.

- The clock now has a calendar and alarm capability. Click on the front panel clock for a calendar. Alarms may be set to play a sound, execute a program, and/or scroll a message across the title bar of a window. Alarms can repeat hourly, daily, weekly, weekdays, weekends, fortnightly, monthly, quarterly and yearly. An alarm that is delayed (because ProgramBar was not running when it was triggered) may be ignored, accumulated, deleted or set to trigger.

- The live desktop may now store documents as icons. These files are stored as a link to the document and take up negligible system resources. Double clicking on the icon will run the associated application and load the document. The title of the document may be changed to a more appropriate long title. This is similar to the long file name system employed in Windows95. Documents are dragged from File Manager onto the desktop, or added using the desktop popup menu.

  A Document may be duplicated from its system menu or by pressing the CTRL key while selecting and moving the document icon.

- Icons on the desktop may be separated into document and program types. Each type may be placed on a different edge of the screen. It is possible for the desktop to automatically arrange itself.

- The groups present in Program Manager may now be rescanned while ProgramBar is running. This option is available from the configuration dialog.

- The configuration dialog may now be minimized. It does not (and never has) appeared on the front panel application selection. Use the Start|Configuration|ProgramBar to find the window again.

- Documents on the desktop may be saved individually or in groups to `layout' files. These layout files may appended to or override the desktop layout saved between sessions.

- The Find Files dialog has been converted to a modeless dialog, in the same fashion as the configuration dialog. It may be left permanently on the desktop if required, rather than locking up ProgramBar.

- ProgramBar no longer appears on the Alt-TAB list when permanently visible. Also it has been removed from the DOS box Alt-TAB list.

- Intermittent interference with the activation of a screen saver. Fixed.

- Restructured help file for better readability and to overcome a limitation of the Help compiler (HC31 v3.10.505).

- Occasional truncation of the number of Program Manager groups (due to a discrepancy in the Program Manager INI file) fixed.

- Removed remaining memory leaks using Borland CodeGuard.

- Several controls in the configuration dialog that are disabled when the front panel is locked were freed again under certain circumstances. This protection ensures that ProgramBar is left in a safe state. Fixed.

- Environment variables used as arguments to programs are now parsed and replaced. Literal substitution of % for %% also carried out. Available for Program Manager groups, the fast access menus, alarms, and in the **Start|Run...** dialog.

- Left and right mouse clicking over a disabled window on the front panel no longer produces the system menu or close application menu.

- When run as the shell application, ProgramBar failed to close down Windows when it is was closed itself. Fixed.

- StartUp group now executed when ProgramBar is the shell.

- Visible flyby hints failed to update with the new window text. Fixed.

- History lists in **Run...** and **Find Files...** dialogs now saved between sessions.

- Bug in font selection dialog box, failure to map font point sizes too far apart (e.g. 14pt to 10pt). Fixed.

Version 2.11, released 1-Dec-95

Version information

- 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 (i.e. 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 ellipsis (...) 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 sub directories.

- 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 wild cards 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.

- Intermittent 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, released 21-Nov-95

Version information

- 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

Version information

- 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, released 23-Oct-95

Version information

- 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

Version information

- 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

Version information

- First public release.

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!).

Identifying applications for task switching

Using ProgramBar Known problems Troubleshooting problems Reporting bugs

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 know 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, or a minimized flag (i.e. always has an iconic presence on the desktop).
- A document link cannot be an application.
- If the top level window has zero size then it cannot be an application.
- If both the window class and module name are on the exclusion list then it cannot be treated as an application.

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 DOS windows

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 have spent some time investigating options and, short of writing my own DOS emulator, cannot solve these problems. The resource loss experienced can lead to a catastrophic failure of Windows or accumulate over several Windows sessions and degrade performance. Clearing the problem requires a soft reboot (Ctrl-Alt-Del).

Finding application icons

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 v2.0 compatibility

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 to two to co-exists reasonably peaceably...

- Dashboard allows access to extended screens though 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.

From version 2.20 onward I am no longer working on problems associated with Dashboard 2.0. Starfish Software have brought out a later version and I was generally unhappy with the amount of memory used. An 8MB machine needs all the memory it can get while compiling...

Virtual desktops

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.

I have so far received several responses regarding this problem. ProgramBar appeared at the bottom edge of the virtual desktop for the graphics card in question. I have decided that this is not a problem that is worth trying to solve as it only applies to a relatively small number of graphics cards.

Providing a keyboard interface

ProgramBar is designed to be as unobtrusive as possible, sitting on your desktop hidden away waiting to be activated by a mouse movement. Several people have requested a keyboard interface, mostly because they don't want to pick up the mouse when they have just finished typing.

A future version of ProgramBar may contain a keyboard shortcut that activates ProgramBar, displaying the Start menu so that the cursor keys may be used to navigate the menus. Unfortunately I cannot see that a more detailed interface is possible or consistent with the unobtrusive nature of ProgramBar.

Subclassing windows

What is window subclassing?
Each window on your desktop is associated with an application or DLL. Windows dispatches messages to the window each time an action is required or a change has been made that an application might consider important. For example, each movement of the mouse over a window results in a message being sent to the window informing it of the new position of the mouse.

By default each window stores an entry point (known as the window function) to the owning application which is expected to process the message. Not all windows belonging to an application need to use the same window function.

Window subclassing is the act of substituting the window function for a given window with a window function in the same or another application. This does not change the ownership of the window. Rather it allows another application to listen in on the messages being received by a window, changing them if desired. It is necessary that the default window function is stored so that the message can then be passed on to the original owner for the correct response.

What is the problem with subclassing?
The problem arises when more than one application wants to subclass the same window. There is no method provided for chaining these substitutions, it is up to the application that performs the substitution to replace the original window function when it is done.

Imagine the following situation: Application X and then application Y both subclass the same window (owned by an application Z). Both X and Y store the window function they discover is associated with the window, Y sees the window function placed there by X and stores it away safely. Application X is now closed, it has to undo its changes. It can either replace the original window function (preventing Y from working) or leave the window function unchanged. Now if Y is closed then it also has to restore the original state, either restoring an invalid window function to X (and crashing Windows) or restore the default window function (which can be obtained by other means).

The above example illustrates that window subclassing is competitive rather than co-operative. If an application gets it wrong then Windows can fall over. The problem is compounded when 3 applications try to subclass the same window.

How ProgramBar subclasses a window
ProgramBar only subclasses one window at a time. This minimizes on the system overhead incurred when monitoring system messages. In attempting to make the best of the competitive subclassing, ProgramBar employs the following strategy in restoring the window function:

- Checks whether the window function it will restore is still valid. If it is not then the default window function is restored. This is the worst possible situation, another application subclassing the window may be bypassed.

- If the current window function is the one that ProgramBar has set, it will safely restore the previous window function. The situation has been restored to that before ProgramBar subclassed the window.

- If the current window function is different to the one that ProgramBar placed there then nothing is done.

Another application has subclassed the window, it should be left alone.

- Should ProgramBar's subclassing function be called for a window other than the currently subclassed window then the default window function is restored to that window.   Another application has incorrectly attempted to restore the ProgramBar window function. This situation is also undesirable, another application subclassing the window may be bypassed.

ProgramBar is attempting to avoid compromising the system but cannot be held responsible for the behaviour of other applications.

When designing Windows, subclassing should have been implemented in the same way as system hooks: a linked chain controlled by Windows rather than competitively by applications running on the system.   An application can then request that a link is removed, Windows being responsible for maintaining the integrity of the chain.

Using ProgramBar as the shell

You can successfully use ProgramBar as the shell application, however there are several limitations. A full implementation of a shell application should have the following features:

1. May be placed on the line **shell=** in `system.ini`.

2. Will run the same StartUp group as Program Manager when run as the shell. Program Manager only runs one group called StartUp, even if there are several groups of the same name.

3. Exit Windows when closed itself, allowing applications to save data files before exiting.

4. Provide DDE replacement for Program Manager. This includes DDE requests for icon images, names of groups etc.

5. Allow addition/removal of items from Program Manager groups. Also allow the ordering of items in the groups.

6. Allow security: disabling certain features, password protecting others (optional)

The current implementation of ProgramBar provides 1,2 and 3 in the above list. Microsoft documentation states that part of item 4 is required to support DOS applications by providing the display icon, working directory and description.

Items 4,5 and 6 are planned for a future release of ProgramBar.

What is Shareware?

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 often 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.

Registration details

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 hard cover book, that is only one copy may be running on one machine at any one time.

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 £15 (fifteen British pounds sterling). Cheque, postal order, or international 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 licensed 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)
First user	15
2nd to 5th user	12
6th to 10th user	10
11th to 25th user	6
26th to 50th user	4

51st user and over	2
Unlimited users at one site	500

The price is capped at approximately 150 users to £500.

An order form is provided for your convenience.

Obtaining the most recent version

Contacting the author   Version history   Overview

ProgramBar's homepage
ProgramBar now has it's own Web site: `http://www.eudoxus.demon.co.uk`. Due to the limitations imposed on the site it is not practical for shareware files to be stored. Pointers will however be provided to the many sites that do contain a copy of the latest version of ProgramBar.

Version naming conventions
ProgramBar is archived at several sites. The file naming convention for v2.1 and earlier was `prgbar??.zip` where `??` represents the version number. From v2.11 onward the file naming convention became `prgbr???.zip`, where `???` represents the version number.

A complete revision of the scope and functionality of ProgramBar will increase the version number by **1.0**. A major revision of ProgramBar with enhanced functionality and feature set will increase the version number by **0.1**. A bug-fix release and minor feature enhancement will increase the version number by **0.01**.

FTP search engines
For a local copy I would recommend visiting an FTP search engine. A search engine may be several weeks behind the contents of archive sites, however. The following is a good FTP search engine that I have used regularly and recommended in the past:

`http://ftpsearch.ntnu.no/ftpsearch`

Shareware archives
Estimated URL's are provided for each site, but the actual location of ProgramBar is at the site maintainers discretion. Most sites provide a search engine, the key word ProgramBar or filename `prgbr230.zip` should be sufficient to locate the file.

SimTel.Net
SimTel is a software repository that uses its archives to build CD-ROMS with appropriate software libraries. Run by Keith Petersen, it is fully supported by Walnut Creek CD-ROM.

Web browser:	`http://www.SimTel.net/`
Directory:	`pub/simtelnet/win3/desktop`
File:	`prgbr230.zip`
URL:	`http://www.SimTel.net/pub/simtelnet/win3/desktop/prgbr230.zip`

Winsock-l
FTP site with many mirror sites world-wide. While not a Winsock application, the site does also archive tools and utilities that may be considered useful to their subscribers.

FTP or Web browser:	`papa.indstate.edu`
Directory:	`/winsock-l/ToolBars`
File:	`prgbr230.zip`
URL:	`ftp://papa.indstate.edu/winsock-l/ToolBars/prgbr230.zip`

Winsite (formerly Cica)
A huge site that has well... huge amounts of software. FTP access through `ftp.winsite.com` shows individual directories tend to be overfilled with files, very hard to find something that you want or need unless you know the filename or are good at guessing from 8 letters. The site now has an index search engine, and the web interface is recommended over the FTP interface.

Web browser:	`www.winsite.com`
File:	`prgbr230.zip`
URL:	`http://www.winsite.com/info/pc/win3/desktop/prgbr230.zip/`

Hensa academic archive
This archive site is primarily setup to serve the British academic community. It is always open to sites in the `ac.uk` domain, and for limited periods from other sites.

FTP or Web browser:	`micros.hensa.ac.uk`
Directory:	`micros/ibmpc/win/j/j099`
File:	`prgbr230.zip`
URL equivalent:	`http://micros.hensa.ac.uk/micros/ibmpc/win/j/j099/prgbr230.zip`

Contacting the author

Registration Overview

Any feedback regarding ProgramBar is welcome, as are suggestions and improvements. Many of the features of ProgramBar have been driven by user request, making ProgramBar the software utility it is today!

ProgramBar now has it's own web page: `http://www.eudoxus.demon.co.uk`. It contains information on the latest version, bug reports, information on the latest version - drop in sometime!

Postal address

Ian Jefferies
24 Meredith Close
Pinner
Middx
HA5 4RP
ENGLAND

Telephone

Postal address: England (0)181-428-1466

E-mail

`ProgramBar@eudoxus.demon.co.uk`

What makes a good book?
Well, that's 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 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.

Registration form for ProgramBar

Registration Contacting the author Overview

The 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.

Single user registration form
Site license registration form

ProgramBar v2.30 single user registration form.

Mail to:

Ian Jefferies	Telephone:	(0)181-428-1466
24 Meredith Close	E-mail:	`ProgramBar@eudoxus.demon.co.uk`
Pinner Middx		
HA5 4RP		
ENGLAND		

Your details:

Name:	____________________
Company position:	____________________
Contact address:	____________________
	____________________
	____________________
	____________________
E-mail address:	____________________
Day telephone number:	____________________

Purchasing details:

	Quantity:	**Price per unit**	**Total:**
ProgramBar v2.30	__________	@ £15 each (fifteen British pounds)	__________

How did you hear about ProgramBar?

Further comments:

ProgramBar v2.30 Site license registration form.

Mail to:

Ian Jefferies	Telephone:	(0)181-428-1466
24 Meredith Close	E-mail:	ProgramBar@eudoxus.demon.co.uk
Pinner Middx		
HA5 4RP		
ENGLAND		

Your details:

Name:	______________________________
Company position:	______________________________
Contact address:	______________________________
	______________________________
	______________________________
	______________________________
E-mail address:	______________________________
Day telephone number:	______________________________

Purchasing details:

Site license:	**Payment in £ (British pounds)**	**Total:**
2 to 5 users	1 user (£15) + £12 / extra user	__________
6 to 10 users	5 users (£63) + £10 / extra user	__________
11 to 25 users	10 users (£113) + £6 / extra user	__________
26 to 50 users	25 users (£203) + £4 / extra user	__________
51 users and over	50 users (£303) + £2 / extra user	__________
Unlimited users at one site	£500	__________

How did you hear about ProgramBar?

Further comments:

Acknowledgements

Registration Contacting the author

A great many people have written to me with ideas, suggestions and bug reports. Their help and support have been invaluable in helping me continue with the development of ProgramBar.

The beta testers
There are quite a few people who have helped make ProgramBar what it is today, and I would like to take this opportunity to thank them for all the hard work that they have put in. They have suffered with buggy code and put in a great many hours of testing to help find and remove these problems. Their countless sensible suggestions have improved ProgramBar, and they have been extremely patient as deadline after deadline has slipped.

ProgramBar would not be as good as it is without these people, if you like ProgramBar then please drop them (and me!) an e-mail thanking them for the effort. As always, any outstanding bugs and problems are for me to handle - please don't hassle them as I've done enough of that already!

In no particular order:

Simon Thomas	`<ecuapita@oznet02.ozemail.com.au>`
Helen Orme	`<horme@scherer.co.uk>`
Vagn K. Poulsen	`<vkp@sigma.ou.dk>`
Stephen Kitt	`<s.kitt@ed.ac.uk>`
Matthew Armsby	`<mattheway@tcns.co.uk>`
Jeff Seiden	`<seidenj@ix.netcom.com>`
Ross Fitkin	`<ross.fitkin@mailbox.swipnet.se>`

My father
Decent personal computers were only just coming onto the market when I was eleven, and I was already convinced that I wanted to be involved with them. My father was working in the pharmaceutical industry and could see the impact that they were going to have in the future: they took over many of the tedious calculations that had previously taken hours or even days to perform, managed data, and opened up new areas of research and data processing previously unobtainable.

Even though he did not want to adapt to use them - he retired before their use became too widespread - he had the foresight to give me the opportunity of learning how to use these new tools. He invested in my first computer: a Sharp MZ-80K with its 48K of RAM, tape drive and built-in monitor... and I loved it!

Later, with his blessing, I bought my second computer: a BBC micro. He listened with great patience as I told him of all the things that I was doing with it. Many of them I don't think he actually understood, but that wasn't important as he knew that I had found something that I could do, and do well.

Just over 4 years ago I purchased a 486/33 and Borland C++ version 3.1 so that I could teach myself serious computer programming on a serious machine. My father could now do nothing to help me, but he still listened as I told him of the things that I had done. I told him of ProgramBar and how widespread it had become though the Internet, and I could see that he knew the investment he had made all those years ago had been worth it.

On the 25th January 1997, my father died suddenly and unexpectedly of a heart attack at the age of 69. Although he will not see the things that I will do in the rest of my life, I know that I will have the opportunity to do them because he believed in me.

I dedicate ProgramBar to the memory of my father, who's foresight made it possible.