

Toolbar

1.11

Juan M. Aguirregabiria
Fisika Teorikoa. Zientzi Fakultatea
Euskal Herriko Unibertsitatea
P.K. 644, 48080 Bilbo (Spain)
Internet address: wtpagagj@lg.ehu.es

June 1993

Copyright

The program **TOOLBAR.EXE**, the help file **TOOLBAR.HLP** and this documentation file (**TOOLBAR.WRI**) are copyrighted:

Copyright © Juan M. Aguirregabiria 1993
All rights reserved

Other brand and product names are trademarks or registered trademarks of their respective holders.

You are granted the right to freely use and distribute *Toolbar* for *non-commercial* use, as long as the aforementioned three files are held together and unchanged. You have to pay nothing for using the program, but if you like it you could consider spending some time in improving the spelling and writing in this documentation and in the help file. Since English is only a second language for me (well, you are right: only a fourth language), I would specially acknowledge any help in this job. Your work would be duly credited and you would receive future program updates as fast as possible.

Disclaimer

In no event will the Author be liable to users for any damages, including but not limited to any lost profits, lost savings or other incidental or consequential damages arising out of the use or the inability to use this program, even if the Author has been advised of the possibility of such damages, or for any claim by other party.

Hardware and software requirements

To run *Toolbar* you need Microsoft Windows 3.1 and the hardware necessary to run it.

For spanish-speaking users

Existe una versión castellana del programa y del fichero de ayuda (pero, por el momento, no de esta documentación) que puede obtenerse directamente del autor.

New in version 1.11

This version is to correct a nasty bug that gave a **Not enough memory** error when trying to add an element to a free toolbar. I apologize for the inconveniences.

New in version 1.10

- The history list feature has been added to the **Run . . .** dialog box.
- You can use **Advanced options** in the **Program** command line. This lets you control the way in which dropped files are passed to a tool. You can even prompt the user before starting the tool.
- You have more control over the directory in which a tool will start. *Toolbar* will use **Directory** if available, the first dropped file's directory or the program's directory if anything else fails.
- A toolbar can now be hidden (instead of minimized or quit) when one of its tools is used.
- When using *Toolbar* as shell, the form **win program** can be used to have **program** automatically started.
- Some minor bugs leading to memory leaks have been fixed.
- More specific error messages are now issued when a tool cannot be started.
- The pointer appearance when changing elements order by dragging looks now much better. In fact, true drag-and-drop is now used.
- One can now drag elements onto and **from** *Toolbar*. This is specially handy when copying elements from a toolbar to another.
- *Toolbar* can now use any *Program Manager's* group file as **INI** file. It will even translate it to *Toolbar's* format.
- *Toolbar's* caption is now updated only if something has changed. This minimizes the potential flicker of some underlying windows.

Acknowledgments

I want to acknowledge very stimulant feedback from my brother Javi and some people from *Internet*: Geoff Bishel, Olaf Winkelhake, Ronnie (Rocker), Josu Oyanguren, Frank Robijn, Michael John Raine and Tian (DARSTELLUNG).

Introduction

Toolbar is intended to provide some features that I found lacking in the *Program Manager* and *File Manager* of the retail Windows 3.1. To start with, while working in *File Manager*, you can drag an item and drop it over some running programs. But if one of these programs is not running and you drop the item over its icon, the program is not started. Instead, the dropped item is added to the corresponding group of *Program manager*. Of course, sometimes this is just what you expected. One could say that the groups in *Program manager* are devoted more to organize programs than to run them. On the other hand, most programs still lack the ability to accept dropped files and you cannot use with them this drag and drop feature of *Program Manager*.

Since my screen tends to be cluttered with windows, I also wanted the possibility to locate (in a permanent and compact way) at a corner of the screen the utilities I use most often. *Program Manager* groups are nice but big and I do not need to see the names of my preferred utilities: I am able to recognize them by their icons. The utilities should be always at hand ready to be started by double-clicking at them (as in *Program Manager*) or by dropping an item on them. For instance, this is a nice way to provide the most important facility missing in *File Manager*: a file viewer.

Toolbar is the answer to these wishes. It provides an alternative to the groups of *Program Manager* with the following distinctive characteristics:

- It can be used either as a substitute for *Program Manager* (which, in turn, can be run from *Toolbar*), or as a normal application installed in *Program Manager* (or in the current shell program).
- It is more oriented to run programs than to organize them. There are more ways to run programs and more advanced settings can be used.
- All programs in a toolbar are always visible (as long as the whole window is on the screen). They are held together in a very compact way, to be as unobtrusive as possible. Optionally, you can force a toolbar to be always visible on top of other windows. In order to save space, even the title bar can be used to have at hand useful information.
- Items from *File Manager* can be dropped virtually on any program installed as a tool. It only needs to be able to accept file names in the command line. Most MS-DOS and Windows programs have this possibility. A program can also be started by double-clicking at it or by using the **Enter** key or a user selected accelerator key.
- You can choose among one instance of the program for each dropped file or, instead, a single instance to deal with all the dropped files.
- Programs can be easily added, removed and reordered. The corresponding caption, command line, working directory, icon and other options can be easily changed.
- You can implement a full tree of nested toolbars.
- Unlike *Program Manager*, *Toolbar* is able to accept and **provide** dragged files.
 - Apart from these distinctive characteristics, *Toolbar* works much in the same way than *Program Manager*.
- In fact, *Toolbar* is able to use (and to translate to its own format) group files from *Program Manager*.

Note

After having completed version 1.00 of *Toolbar*, I received my CD-ROM containing the CICA collection of Microsoft Windows programs. As I suspected, there exist other programs that provide similar capabilities. Sincerely, I do not know them but the idea was rather obvious. In any case, *Toolbar* might have an advantage: it is completely free for non-commercial use!

Installing *Toolbar*

To install *Toolbar*:

- Copy to the desired directory the executable file, **TOOLBAR.EXE**, the help file **TOOLBAR.HLP** and, optionally, this documentation file: **TOOLBAR.WRI**. When you execute it, *Toolbar* will install, in the same directory, **TOOLBAR.INI** or the corresponding **INI** file.
- Install the program in a Windows group. You may want to install it in the **StartUp** group to have *Toolbar* automatically started at the beginning of each Windows session.
- You can use the dialog box of **Properties** (in *Program Manager*) to append to the command line the name of an **INI** file to be used instead of the default **TOOLBAR.INI**. In this way you can install and run simultaneously different copies of *Toolbar*. You can even install one or several trees of nested toolbars.

Running *Toolbar*

You can run *Toolbar* exactly in the same way as any other Windows program: by double-clicking on it, by using the **Run** menu entry, etc. If you find yourself using *Toolbar* on every Windows session, you could install it in the **StartUp** group to have *Toolbar* automatically started each time Windows starts.

The first time you run the program it will create the **TOOLBAR.INI** (or the **INI** file you selected in the installation process). In this file the program will automatically save information on the window location and appearance and on the installed tools. In this way, the next time you run the program it will start exactly at the same screen location and with the same tools and options.

If the program is running and you try to re-run it, the already running instance will be activated and its size restored if it was minimized. This can help locating the program if it gets lost under other windows. You can only have a running instance of *Toolbar* for each **INI** file, because I cannot think of any advantage of having two exact copies of the program running at the same time. But, of course, by using different **INI** files, as described in the Installation section, you can run simultaneously different toolbars.

Using nested toolbars

You can use these features to implement a full tree of toolbars! Simply install in a toolbar (which, to be always visible and at hand, would probably have the attribute **Top window** set; see below) one or several secondary toolbars (each with a different **INI** file and, probably, with the attribute **Minimize**, **Hide** or **Quit** set). This can be nested to any depth to give a very complete and yet unobtrusive way of accessing your tools. You could even use *Toolbar* as the Windows shell by checking the **Use as shell** command (see below).

Installing elements (tools and toolbars)

To add an item (a tool or a secondary toolbar) to the current toolbar use the **New element...** entry in the system menu of *Toolbar*, or press the **Ins** key. (Alternatively, if there is some piece of *Toolbar* window with no tool, you can double-click there.) After having selected the type of element in the **New** dialog box, the **Tool properties** or **Toolbar properties** dialog box will appear to let you define the new tool. You can also drag a file from the *Program Manager* and drop it on the title bar of *Toolbar*. In this case, the **Tool properties** dialog box will appear with default values for the **Title** and **Program** entries.

In any case, if you exit the **Tool properties** or **Toolbar properties** dialog box with **OK**, the new tool will be installed and the *Toolbar* window will be resized to display it. You can change the appearance of this window by using the **Options...** menu entry.

You can also drag an element from a toolbar and drop it on another toolbar. It will be copied there and no dialog box will be opened.

Selecting an element

To select a tool you click once at it with the mouse or use arrow keys to change the selection. When a tool is selected you can change its properties with **Alt+Enter**, delete it with **Del** and run it by pressing the **Enter** key (or **Shift+Enter** if you want it to start minimized).

Changing element properties

To change the properties of an element, select it as described above and use the **Properties...** entry in the system menu, or the **Alt+Enter** key combination. The **Tool properties** or **Toolbar properties** dialog box will appear to let

you customize the element.

Tool properties and Toolbar properties dialog boxes

This dialog box is started when installing a new element (by using the **New element. . .** entry in the system menu, or the **Ins** key) or customizing the selected element (by using the **Properties. . .** entry in the system menu, or the **Alt+Enter** key combination). Apart from the self-explanatory **OK**, **Cancel** and **Help** buttons, you can use the following items to change the attributes of the element:

Title: The caption that will optionally be displayed to identify the element. **If you do not provide any title, the element will remain installed only during the current session of *Toolbar!*** (This can be useful to install an element that will be used only during the current session.) To display the title in two lines, use the pair **\n** as line break.

Program: This entry appears only in the *Tool properties* dialog box and contains the complete file specification of the program to be run, including optional parameters. You can enter directly the program specification or search for it through your disks and directory tree by using the **Browse. . .** button and the **Select program** dialog box that will be displayed. Instead of an executable file you can also specify here any (document) file whose extension is associated by the Program Manager to an executable file. For instance, using file.wri you will actually get write file.wri; i.e., the write.exe program with the file.wri file as parameter.

File: This entry appears only in the *Toolbar properties* dialog box and contains the complete file specification of the **INI** file that will contain options and tools for the nested toolbar. You can enter directly the file specification or search for it through your disks and directory tree by using the **Browse. . .** button and the **Select INI file** dialog box that will be displayed.

Directory: The directory in which the program will be started. Though this entry is optional you should consider filling it, because many programs will only work properly if they start from the right directory (most often their own directory). If this entry is not used, the program will start in the directory of the first dropped file or, if no file has been dropped, in the program's directory.

Icon: This entry specifies the icon that will represent the element in the toolbar. By default, the first icon in the executable file will be used, but you can select another icon by using the **Change. . .** button and the **Select Icon** dialog box that will be displayed.

Minimized: If the entry is checked, the tool will start minimized. If it has not been checked, you can still run the tool minimized if the **Shift** key is held down when the program is invoked.

Multiple: This entry controls the behavior of the tool when several files are simultaneously dropped on it. If the entry is checked, each dropped file will start its own instance of the program. If it is not checked, all the dropped files will be appended to the command line of a single instance of the program. Note that whatever this setting is, if different files are dropped at different times, each of them will get its own copy of the tool. Remember also that MS-DOS programs have very

short command lines (less than 128 characters). As a consequence, they can receive only some of the dropped files and the last received file can have its file name truncated. Nothing of this can happen if this entry is not checked.

Fast load key: This optional entry lets you select an accelerator key that will start the element.

Advanced options: This entry lets you specify if you want to use advanced options in the **Program** command line. See **Using advanced options to run programs**.

Exactly as happens in *Program Manager*, when an MS-DOS application is minimized the icon selected in this dialog box will be used, but Windows applications have their own icon displayed. However, there is an exception when using a toolbar that has been installed as element of another toolbar. Though it is a Windows program, if an icon has been selected in the aforementioned **Icon** entry, it will be used instead of *Toolbar's* icon and it will be displayed with a special look: as a button. A primary toolbar (i.e., a toolbar that has not been installed inside another toolbar) will use *Toolbar's* icon, but a power user can change this by adding to the corresponding **INI** file two lines in the form:

```
Icon file=file
Icon number=n
```

where *file* is the complete specification of the icon file and *n* is the number of the icon in the file starting at 0.

Select icon dialog box

This dialog box is started by using the **Change . . .** button in the **Tool properties** or **Toolbar properties** dialog box. Apart from the self-explanatory **OK**, **Cancel** and **Help** buttons, you can use the following items to change the tool icon:

File name: The complete file specification of the file containing the icon. The default is the program file but you can enter directly a new file specification or search it through your disks and directory tree by using the **Browse . . .** button and the **Select icon file** dialog box that will be displayed.

Current icon: This list box lets you select the icon to be associated with the element. You can also select to have no icon.

Select program, Select INI file and Select icon file dialog boxes

These dialog boxes are invoked but not provided by *Toolbar*. They are three instances of the common dialog box that most Windows 3.1 programs use to find and open a file. You are surely familiar with it and you can find a description in your Windows manual.

Changing elements order

To change the position of an element, pick it with the mouse and holding down the left button move the pointer (whose appearance will change to show the operation in course) over the desired place, then release the mouse button.

Removing an element

To remove an element, select it as described above and use the **Remove** entry in the system menu, or the **Del** key. You will be prompted to confirm the deletion.

Running an element

To run an element, you simple use the mouse to double-click at it. You can also select it as described above and use the **Enter** key. The third way to start an element is by using the **Fast load key** selected in the **Tool properties** or **Toolbar properties** dialog box.

The program will start minimized if you have chosen it as the default behavior in the **Tool properties** or **Toolbar properties** dialog box or if the **Shift** key was held down when the program was invoked.

To start a tool with a data file, you drag the latter from the *Program Manager* and drop it over the tool.

Changing program options

To change *Toolbar* options use the **Options . . .** entry in the system menu, or the **Ctrl+O** key combination. The **Toolbar options** dialog box will appear.

Using advanced options to run programs

If you check the **Advanced options** entry in **Tool properties**, before starting the tool *Toolbar* will scan the **Program** command line (after having appended eventual dropped files) to seek the following special options, *which are not case sensitive*:

If the **Multiple** instance option is checked, each dropped file will obtain its own copy of the the tool and you *must* control the exact location and format in which the file is passed to the tool by using one or several of the following options:

%P will be replaced by the complete file specification, including drive, directory, name and extension.

%F will be replaced by the file specification with the extension (and the corresponding period) removed; i.e., it will include drive, directory and name.

%D will be replaced by the file drive and directory. The trailing \ will be retained only in the case of the root directory.

%N will be replaced by the file name, excluding drive, directory and extension.

%E will be replaced by the file extension (without the starting period).

If the **Multiple** instance option is not checked, all dropped files will be appended with their complete file specification after the tool program. Nevertheless, you can place some options after all these files by using the following option:

%A The remaining of the command line will be appended after eventual dropped files.

This is useful when the program needs options after file names, because by default dropped file names are appended to the end of the **Program** command line. It can appear only once.

All other % options will be deleted. To have fixed options in this form,

%% will be replaced by a single %. This is useful if your program needs its own % options.

Furthermore, you can prompt the user for additional information:

%Q ***A Toolbar is about to execute dialog box*** will be opened to let the user introduce additional options or change the current ones. This is useful when the program needs options that cannot be provided by drag and drop. It can appear only once and the command line will not be further scanned after this option. For instance, if you want to drop files over an archiving program called **MyArc**, you could use something like **MyArc %Q** to be prompted for the archive file name where the dropped filename must be added.

Toolbar is about to execute dialog box

This dialog box will appear if the **Advanced options** is checked and the **%Q** option has been included in the **Program** command line. You will be prompted with the actual command line that will be executed (including eventual dropped files). The cursor will be at the position indicated by **%Q**. You can then edit the line to add or change options. If **OK** is pressed the command line will be immediately executed. You can also choose **Cancel** to abort the procedure and **Help** to obtain help.

Running a program

You can run from *Toolbar* any DOS or Windows program (and not just the installed tools as described in **Running an element**). Simply use the **Run...** entry in the system menu or press **Shift+R**. The **Run** dialog box will be displayed.

Run dialog box

This dialog box is started by using the **Run...** entry in the system menu, or the **Ctrl+R** key combination. Apart from the self-explanatory **OK**, **Cancel** and **Help** buttons, you can use the following items to customize the toolbar:

Command line: Select here the complete specification of the executable file, including optional parameters. You can enter directly this information or to search for the program through disks and directories by using the **Browse...** button and the **Select program file** dialog box that will be displayed. Instead of an executable file you can also specify here any (document) file whose extension is associated by the Program Manager to an executable file. For instance, using file.wri you will actually get write file.wri; i.e., the write.exe program with the file.wri file as parameter. This entry contains also a list of the last ten commands that can be reissued simply and will even be available in the next session if **Options** is On.

Run minimized: Check this entry if you want the program to start minimized as an icon.

Save command in history list: Uncheck this entry if you do not want the program saved in the commands history list.

Toolbar options dialog box

This dialog box is started by using the **Options...** entry in the system menu, or the **Ctrl+O** key combination. Apart from the self-explanatory **OK**, **Cancel** and **Help** buttons, you can use the following items to customize the toolbar:

Rows: Select here the number of rows used to display the tools. **Columns** will be automatically changed to reserve space to display all the tools.

Columns: Select here the number of columns used to display the tools. **Rows** will be automatically changed to reserve space to display all the tools.

Frame: Select here the width, in pixels, of the frame to be drawn around each element. It can be 0 to display no frame and from 1 to 9 to display it with different widths.

Button like: If this entry is checked, *Toolbar* will display tool icons in the form of standard buttons. But, remember that they are not buttons; for instance, to activate one of them you must double-click at it. (Clicking once will only select it.)

Small font: If this entry is checked, *Toolbar* will use a small but very readable

font to display element captions (this allows seeing larger titles). If it is unchecked, it will use the same font as *Program Manager*.

Titles: If this entry is checked, *Toolbar* will display a caption under each element. If it is unchecked, captions will not be displayed to save space, except for the elements without icon, which will have the caption displayed instead.

Top window: If this entry is checked, the *Toolbar* window will be always visible over other windows on the screen (well, at least over other not top windows). If it is unchecked, the window can be covered by the active window in the usual way. Note that the property of being a top window is hereditary in Windows: child windows will have it and will be displayed over other windows. This can be a little inconvenient when displaying a help window.

Caption: In this entry you can select the caption to be displayed in the title bar of *Toolbar*. You can select here an arbitrary string, but the following pairs of characters will have a special meaning and can be used to save space by using the title bar to display useful information:

Char pair	will be replaced by
%d	the current day (one or two digits)
%D	the current day (always two digits)
%m	the current month (one or two digits)
%M	the current month (always two digits)
%y	the current year (one or two digits)
%Y	the current year (always two digits)
%h	the current hour (one or two digits)
%H	the current hour (always two digits)
%i	the current minute (one or two digits)
%I	the current minute (always two digits)
%s	the current second (one or two digits)
%S	the current second (always two digits)
%f	the free memory in Mb
%k	the free memory in Kb
%b	the free memory in bytes
%r	the free resources percentage

For instance, **%h:%I - %fMb - %r%** will display a title in the form **1:02 - 3Mb - 71%** indicating the current time, free memory and free resources.

Frequency: In this entry you can select the frequency, in seconds, at which the caption defined in the previous entry is refreshed to display the current values of the above defined information.

After a tool is started *Toolbar* look may change according to the state of the following four radio-buttons:

Do nothing: If this entry is checked, *Toolbar* will not change when one of its elements is activated in either way.

Minimized: If this entry is checked, *Toolbar* will be shown as an icon when one of its elements is activated in either way. This is specially useful when implementing a full tree of toolbars.

Hide: If this entry is checked, *Toolbar* will hide when one of its elements is activated. This also is useful when implementing a full tree of toolbars. To see again

the hidden toolbar you try restarting it. The hidden toolbar will be made visible, so this will be a bit faster than the following possibility. This feature is not available when *Toolbar* is being used as the shell, because the retail *Task Manager* does not show hidden tasks and, thus, using a tool may lead to a situation in which nothing can be done, not even exiting Windows!

Quit: If this entry is checked, *Toolbar* will exit when one of its elements is activated. This also is useful when implementing a full tree of toolbars.

When *Toolbar* ends the installed tools are saved to the **INI** file. Other setting may also be saved according to the state of the following two buttons:

Options: If this entry is checked, *Toolbar* will save the current options defined in this dialog box as the default to be used in the next session.

History: If this entry is checked, *Toolbar* will save the last ten commands, which will be then available in the next session.

Using *Program Manager's* group files

Instead of a normal **INI** file, you can specify a group file from *Program Manager* (they are usually placed in Windows directory and have extension **GRP**). *Toolbar* will do its best to translate it. (To make sure that *Toolbar* chooses the right number of columns and rows, select *Automatic* in the *Options* menu of *Program Manager*.)

To save its configuration in the ordinary way, *Toolbar* will not use the group file, but an **INI** file with the same drive, path and name (but extension **INI**).

The next time the toolbar is started, it will read both the group file (to find the installed tools, the number of **Columns** and **Rows**, and the toolbar **Caption**) and the **INI** file (to find the remaining entries in **Options**).

Note that, in this way, tools that you add to the toolbar or individual **Properties** that you change will not be used in the next session. So, you might prefer to instruct *Toolbar* to use in future sessions instead of the group file the **INI** file that it has automatically created.

This is specially useful after a new program has been installed. Its group file can be readily converted to *Toolbar's* **INI** format.

Using *Toolbar* instead of *Program Manager*

If you really like *Toolbar*, you could try using it as a replacement shell. To do that, simply check in the **Use as shell** entry in the system menu.

To recover *Program Manager* as shell, use again the same command.

In any case, you will be given the chance to immediately restart Windows with the new shell. If you do not accept this (or if Windows does not actually exit because some DOS box is running or some other application refuses to finish) the new shell will be used only in the next Windows session).

Since after the file specification of the Windows shell program no command line can be added, you can only install as shell an instance of *Toolbar* that uses the implicit **TOOLBAR.INI**.

Note also that you can also install *Program Manager* as a tool to be started from *Toolbar*, much in the same manner as you can install *Toolbar* in a *Program Manager* group. Moreover, its toolbars can be nested to any depth, unlike *Program Manager* groups (See **Using nested toolbars**). You may take advantage of the features of both programs.

The auto-start facilities provided by the **StartUp** group of the *Program Manager* appear in *Toolbar* in the form of the **STARTUP.INI** file, whose elements, after having been installed by *Toolbar*, will be automatically run when Windows starts if *Toolbar* is the current shell. The **[load]** and **[run]** entries of **WIN.INI** are also correctly started.

Saving options, commands and tools

To minimize disk access *Toolbar* usually saves the list of installed tools, the current options (if **Options** is On) and the last ten commands (if **History** is On) only when the program exits. If the Windows session is ended abruptly (because some program crashes, for instance) you will lose the changes performed during the current session. To avoid this, you can save at any moment the current configuration without actually exiting by using the same trick that you can use in *Program Manager* and *File Manager*: simply try closing the program while holding down the **Shift** key. In consequence, it is enough to press **Shift+Alt+F4** to have the configuration saved.

Dragging elements

You can drag an element from a toolbar and drop it onto another toolbar (where it will be copied) or, unlike *Program Manager*, onto any other program accepting dropped files. In the latter case the result will depend on the program, of course. Nevertheless, note that some programs (including *Program Manager* and *File Manager*) incorrectly assume that they are the only able to provide dragged files. This will decrease the usefulness of these features with these programs.

Commands

To save space, *Toolbar* does not have its own menu. It uses instead the system menu that can be accessed in the standard ways (typing **Alt+Space** or clicking at the symbol in the upper left corner of the window). After the standard entries (**Restore**, **Move**, etc.) it adds the following entries:

Entry	Meaning	Accelerator key
New element..	Add a element	Ins
Properties...	Change current element attributes	Alt+Enter
Remove	Delete the current element	Del
Options...	Change program preferences	Ctrl+O
Run...	Run a program	Ctrl+R
Use as shell	Use <i>Toolbar</i> instead of <i>Program Manager</i>	
Help	Open help index window	F1
About Toolbar...	Version information	

Error and warning messages

The following messages are issued by *Toolbar* when necessary:

Could not run another instance

A tool refused to start because only one instance of the program (or a necessary library) can be run..

Could not create a window

Improbable error that indicates that a window could not be created (perhaps by lack of memory).

Could not register

Improbable error that happens if it was not possible to add *Toolbar* window class to the Windows database.

Could not run the program

A tool refused to start. Perhaps there was not enough memory or the executable file was missing or incorrect.

Exit Windows?

If your answer is affirmative, Windows session will finish. It is only possible if *Toolbar* is the current shell.

File not found

A tool refused to start because the specified file has not been found.

Incorrect Windows version or mode

A tool refused to start because it was created for another version of Windows or because it cannot be run in the current (standard or enhanced) mode.

Invalid program file

A tool refused to start because the program file was corrupt or was created for another operating environment (OS/2, for instance).

Missing or invalid file

An **INI** file must be indicated for each toolbar installed as a tool inside another toolbar.

No association defined for this extension

A tool refused to start because the specified file has an extension that is not associated to any executable file. Specify the latter or define the association in *File Manager*.

No program defined for this tool

A tool refused to start because you have not specified the file to be run.

No timer available

A kind of resource necessary to run *Toolbar* correctly is exhausted.

Not enough memory

There is not enough memory to pass files to a tool (or to perform other tasks). Try closing other applications.

Path not found

A tool refused to start because the path of the specified file has not been found.

Remove element name?

or

Remove the current element?

Last opportunity before deleting a tool from the toolbar.

Restart Windows now?

If your answer is affirmative, Windows will restart with the new shell.

**This element has no name
It will not be available
in the next Toolbar session**

Warning recalling you what will happen with nameless tools.

**This file has no icon
Use icons in Program Manager?**

If your answer is positive, *Toolbar* will show you the icons in **PROGMAN.EXE**.

**Toolbar has no element
To add an element, press Ins
To get help, F1**

Warning indicating that the current toolbar is void.

**Two elements have now the same name
Only one of them will be available
in the next Toolbar session**

Warning recalling you what will happen if tool names are repeated.

Help

You can access the *Toolbar* help index by using the **Help** entry in the system menu of *Toolbar*, or the **F1** key. From a dialog box the **Help** button gives access to the corresponding help topic.

Suggestions and bugs

The Author would highly appreciate receiving information about any bug or problem found in the program. Suggestions to improve *Toolbar* in future versions are also welcome.

If you experience problems when minimizing non-Windows programs and are using a very low (CGA, EGA) or very high (8514, etc.) resolution card, try adding to the **[Options]** section of the **INI** file an entry in the form:

Use DDE=0