

AppBar 4.0

by
NEVERnever Software

[Why AppBar?](#)

[How to use AppBar?](#)

[AppBar.ini?](#)

[New in AppBar 4.0?](#)

[Where to put it?](#)

[Use it as Shell?](#)

[Drag&Drop?](#)

[Icons?](#)

[Sound?](#)

[QuickLoad?](#)

[How to use the Columns setting?](#)

[Windows requirement?](#)

[Known Bugs?](#)

[History?](#)

[UserWare?](#)

[Thanks?](#)

[Contact?](#)

Why AppBar?

Some time ago I saw AppDock, a NeXT-inspired ButtonBar for Windows. I liked it, but some disadvantages made me not to use it. It was Shareware and made with Visual Basic (so I had to place the large VBRUN100.DLL on my small harddisk), and I thought I could do it better.

How to use AppBar?

AppBar starts programs selected from the **ButtonBar**, gets them to front and closes them as long as they are Windows applications. By simply clicking a button with the left mouse-button you launch programs that are not yet launched, or get a program to front if it is already launched. To indicate the running program, the button goes grey. If you double-click that greyed button, the application is closed. That's all. Almost. The top button provides some system options such as About, Move, Setup, Run, Minimise and Close. The second button provides an onboard 'Exit Windows' option.

AppBar.ini?

As most Windows programs today **AppBar** has its own .INI file, called (you may guess now) **APPBAR.INI**. It looks like this:

[System] (see also [*How to use Columns?*](#), [*Use it as Shell?*](#))

	(min - max, default)
Buttons=8	(1 - 48, 8)
Columns=1	(1 - Buttons, 1)
Left=-1	(-1 - 1280, -1)
Top=-1	(-1 - 1280, -1)
QuickExit=0	(0 - 1, 0)
CloseAll=1	(0 - 1, 1)
DoubleClick=0	(0 - 1, 0)
DefaultIconDir=c:\windows\appbar	(any existing directory)
ShellGroup=startup.grp	(windows 3.1 groupfile, startup.grp)
StayInFront=0	(0 - 1, 0)
Initialized=2	(0 - 2)

[Sounds] (see also [*Sound?*](#))

EnableSound=1	(0 - 1, 0)
SoundDirectory=C:\windows\sound	(any existing directory)
AppBarStart=<none>	(any soundfile, <none>)
AppBarExit=c:\windows\sound\gameover.wav	
WindowsExit=c:\windows\sound\buggle.wav	
ProgramStart=c:\windows\sound\himan.wav	
ProgramClose=c:\windows\sound\thatsall.wav	
DropFile=c:\windows\sound\droopy.wav	
ErrorMessage=c:\windows\sound\glitch.wav	

[Buttons] (see also [*Icons?*](#))

Icon1=c:\windows\appbar\cntrl.ico	(* .icon, *.exe, *.dll filename)
Prog1=control.exe	(any executable filename)
Params=	(any string)
Dir1=c:\temp	(any directory)
Close1=0	(0 - 1, 1)
ShowMode1=0	(0 - 3, 0)
IconNumber1=0	(any number, 0)
ButtonLook1=0	(0 - 1, 1)

Icon2=c:\windows\appbar\winapps.dll

Prog2=c:\windows\winfile.exe

Params2=

Dir2=

Close2=1

ShowMode2=0

IconNumber2=12

ButtonLook2=0

Icon3=c:\windows\appbar\winword.exe
Prog3=d:\winord\winword.exe
Params3=agenda.doc
Dir3=c:\windows\word
Close3=1
ShowMode3=0
IconNumber3=0
ButtonLook3=1

Icon4=c:\windows\appbar\powerpd.ico
Prog4=ppd.exe
Params4=win.ini
Dir4=
Close4=0
ShowMode4=2
IconNumber4=0
ButtonLook4=0

·
·
·

Up to 48 Buttonsettings.

Use **AppSetup** to set up and change you APPBAR.INI file.

New in AppBar 4.0?

In **AppBar 4.0** the following things are new,

- Made use of CTL3D.DLL for 3D looks on dialog boxes.
- Changed clicking mode:
 - mouse_click:** launch program, if NOTALIVE
bring to front, if still ALIVE
 - mouse_doubleclick:** kill program if ALIVE(Note: double_click option is abandoned)
- Fixed ALT-F4 bug: AppBar will exit windows if it is shell and ALT-F4 is pressed.
- Improved the executing of groupfiles.
(Note: ShellGroup now only accepts one groupfile!)
- Added RestartWindows option to ExitWindows Dialog.
- Added simple keyboard interface:
 - escape:** toggles between keyboard interface on/off
 - enter:** is mouse_click on current button
 - delete:** is mouse_doubleclick on current button
 - up arrow:** moves current button 1 up
 - down arrow:** moves current button 1 down
- Added this help file
- Comes with [Tip for AppBar](#) version 1.51.
- AppBar will ask for command line options, if a ? is given as command line parameter.
- Included Version resources in AppBar and AppSetup.

Where to put it?

I don't care where you put the icons, sounds and the APPBAR.EXE file, but the only place I will look for your APPBAR.INI file will be in your Windows directory. Also if you want to use APPSETUP.EXE integrated with AppBar, place it in your path, Windows directory, or in the same directory as AppBar.

Use it as Shell?

Not only will **AppBar** work nicely when you put it in your **load=** line in WIN.INI, but it also works well as shell. Check the checkbox "Use AppBar as Shell" in the system dialogue window of AppSetup to install AppBar as shell. When AppBar is shell, it executes any given program from the WIN.COM command line, the files specified by the **load=** and **run=** lines, and the applications in the groupfiles (separated by a space) specified by the **ShellGroup=** statement in APPBAR.INI. Make sure that the specified groupfiles are created under Windows 3.1. Also will the close button not only close AppBar, but it also exits Windows. If you restart AppBar when it is Shell, it won't start the files from **load=** and **run=** lines or from the **ShellGroup=** any more.

Drag&Drop?

AppBar supports Drag&Drop from Windows 3.1 FileManager. When a file is dropped on a program button, the executable program of that button will be launched with the dropped file as startup parameter. (To be precise, the dropped file will be added to the parameters specified for that button.) When a file is dropped on the two system buttons, the dropped file itself will be launched if it is an executable file, or the associated executable will be launched with the dropped file as startup parameter. When multiple files are dropped on a button, multiple instances of the program will be launched, each with one file from the dropped list.

Icons?

AppBar can use icons from *.ICO, *.EXE or *.DLL files to display icons on the ButtonBar. This means that you can not only use the included icon files (APPICONS.ZIP) and icon libraries (ICONDLLS.ZIP), but also icons from executables themselves, and from Windows 3.1 own MORICONS.DLL file. Use the ButtonLook option to make standard icons look like buttons. IconNumber specifies the number of the icons in a icon library or *.EXE-file (for most *.ICO files it is zero).

Sound?

If enabled, **AppBar** can play a wave-file on six events: AppBar Startup, AppBar Exit, Windows Exit, Program Start, Program Close (by AppBar), Dropping of a File, and Error Message Display. (This feature is only tested with the internal speaker; if you encounter problems with other waveform audio hardware, please let me know.)

QuickLoad?

QuickLoad is an onboard function of **AppBar** to quickly set up the ButtonBar and to copy settings of one button to another. When the QuickLoad button is clicked a QL image is showed on the top button. Now it is possible to drop files from the FileManager on AppBar to load a button. This file will be used for the program and icon settings of that button. No optional parameters or startup directory will be saved, but the 'Close' option will be turned on, and the dropped program will be launched as normal. If a button is clicked (in QuickLoad mode), a special ButtonCopy cursor will be visible. Just click on any other button (except the two system buttons) to copy the settings from the first button. After the QuickLoad button is clicked again AppBar will return to its normal Launch mode.

How to use the Columns setting?

I have received a lot of mail about the use of the Buttons and Columns settings for AppBar. It wasn't always as clear as it should be. I hope to make it clear now. With the Buttons statement you can specify how many buttons you want to have beside the two system buttons that are always visible. The maximum number of Buttons is 48, which would lead to an AppBar strip with 50 buttons (2 system buttons and 48 user-specified buttons).

Now the Columns setting. Here you have to specify how many vertical columns you want to have. So if you like to have an AppBar with 16 user-buttons in two vertical columns, you have to specify these settings:

Buttons = 16,
Columns = 2,

But if you want to have these 16 buttons in one horizontal strip, you would have to specify:

Buttons = 16,
Columns = 18!!,

Because putting 18 buttons in one horizontal row would mean putting each button in a separate column, you need 18 vertical columns. Of course the maximum value of Columns is 50 (with is the maximum of Buttons plus the two system buttons), to have the maximum amount of buttons on one horizontal row.

If you want to have the 16 user-buttons on two horizontal rows, you need the settings:

Buttons = 16,
Columns = 9 (18/2),

Finally, if you want to have these 16 buttons in three horizontal rows, then you can specify

Buttons = 16,
Columns = 7,

AppBar (and that's new for version 3.30) will add three empty buttons, to have 21 buttons on three rows. (You could also have specified **Columns= 8**, which would have lead to $24-18=6$ empty buttons). I hope much is clear now about these settings.

Known Bugs?

The following bugs are known to me:

- The close feature still doesn't work for windowed DOS applications.
- The ShowMode won't work with every program. (especially windowed DOS applications)
- The AppBar move option doesn't work correctly, I am not able to fix this at the moment; if anyone knows a good solution, please mail me.
- Groupfiles will not always be executed correctly when AppBar is used as Shell. If you experience problems with loading groupfiles, try to make a new groupfile from scratch. Groupfiles can get very messy when a lot of settings in it are changed.

Windows requirements?

This version of **AppBar** is written for Windows 3.1 (with the standard MultiMedia extensions). It makes explicit use of MMSYSTEM.DLL, COMMDLG.DLL, SHELL.DLL and CTL3D.DLL.

History?

0.xx tryout versions

3.1 first serious version; added move option, and use of .INI file.

3.14 added QuickView to second button.

3.141 improved fool-proofness; added toggle start/close an application; second button now terminates Windows. (Changed QuickView to ExitWindows).

3.1415 added greyed buttons when a program is active; improved the looks of AppBar; nicer about box. Exit option built-in; made use of Columns possible; increased maximum number of buttons up to 24.

3.1415a Maintenance release; added possibility to specify a startup directory.

3.1415b Maintenance, release; added skipping closing feature, useful when using Windows 3.1.

3.14159

Upgraded to Windows 3.1. Changed redrawing code for icons. Added Setup program. Added Restart Option. Fixed bug in startDir function: current drive is also changed. Added click or double-click mode. Added Start normal, maximised or minimised. Increased maximum buttons to 32. Load= and Run= statements in WIN.INI are supported when AppBar is shell. StayInFront option is added. Changed About box. Improved StayInFront. Skipped close option for all *.COM and *.PIF files. Start AppSetup when AppBar not initialised. Changed .INI file format. Changed the looks of ExitWindows Dialog window.

3.20

Source rewritten for Windows 3.1. Uses ExtractIcon getting the Icon from a file. Displaying with DrawIcon. Added DropFile function. Added Sound. Made Dialog Boxes resolution independent. Fixed dialogue EndPaint bugs. Improved Close function by checking if AppBar is shell. Fixed redraw button problem. Used a timer to check program states regularly. When AppBar is Shell, the Close button will Exit Windows. The groupfile specified by ShellGroup= will be executed when AppBar is Shell.

3.30

Made drop of multiple files possible. If Shell, AppBar won't launch itself when it is also specified in load=, run= or ShellGroup. Use ShellExecute instead of WinExec. Now possible to specify Datafiles and StartUpParameters. Added QuickLoad Option. Max Buttons = 48. Added Run Option. Start ShellGroup programs minimised. Improved closefunction by subclassing closing window, (Thanks to Erkki Riekkola). Bug fixed in max-columns, is now MAXBUTTONS, was MAXAPPS. Added maxsize option. Added save option to move button. Removed IsDosExe(), rely on IsDosWindow() now. Made AppBar restartable when it is Shell. Also, it will execute the options given to WIN.COM when it is shell. Made ShellCheck on "APPBAR.EXE" case insensitive. Use ShellExecute in IsShell() when executing Load= and Run= lines. If Buttons % Rows isn't zero, add Buttons until it is. Added WM_APPBAR_RESTART message. Added minimise button. Added possibility to copy button settings in QuickLoad mode.

UserWare?

This Software is UserWare, no registration, no nothing, just a program to increase the joy of using Windows.

Thanks?

My thanks for testing, debugging, and improving **AppBar** goes to:

Eric A. Rohy, for testing AppBar 3.14159 and 3.20,

Dirk Schumacher and Jürgen Mennecke, for testing AppBar 3.20 and 3.30,

Erkki Riekkola, for testing AppBar 3.20 and 3.30,

Rick Ekle, for testing AppBar 3.30,

Bernt Ribbum, for writing and improving Tip for AppBar.

My beta-testers again, for giving me great ideas, tips, advice

Charles Petzold, for his brilliant book 'Programming Windows',

All 72 users who voted in favour of AppBar as best Windows Shareware and Freeware Application (in the iconbased program launcher category),

All other users, especially those who mailed me telling ne how much they liked or disliked AppBar, or telling the bugs they found.

Martin van der Plas for being a stimulating roommate.

Contact?

geert@ph.tn.tudelft.nl,
and if you want to drink a (Hoegaarden) Beer with me:

Geert van Kempen,
E. du Perronlaan 140,
2624 NA, Delft,
Holland

(You pay the first one, Okay?).

NEVER*never Software 1993.*

mmsystem.dll: DLL for windows MultiMedia functions
commdlg.dll: DLL for common dialog windows such as File Open and Choose Font.
shell.dll: DLL for executing programs and extracting icons from files.
ctl3D.dll: DLL for providing a 3D look to dialog windows. This DLL is not supplied with the retail release of Windows 3.1, but first shipped with excel 4.0, and it now released for public use, supported by Microsoft. Place the latest version of ctl3D.dll in your Windows system directory.

Tip for AppBar is a trashcan utility for AppBar written by Bernt Ribbum. Tip has some special built in functions to work together with AppBar so that its button on the AppBar ButtonBar will change according to the current state of Tip. *See the Tip helpfile for more information.*

APPSETUP.EXE is a standalone setup program for AppBar. It will interactively change the settings in the APPBAR.INI file. AppSetup can be launched from AppBar, and AppSetup can restart AppBar automatically if settings are changed. *See the AppSetup helpfile for more information.*

