AppBar

3.30

by
NEVER*never* *Software 1992.*

Why?

Some time ago I saw AppDock, a NeXT alike buttonbar for Windows. I liked it, but some disadvantages made me not to use it. It was Shareware and made with VisualBasic. (so I had to place the large vbrun100.dll on my small harddisk)., and because I thought I could do it better.

How?

AppBar does basically the same as AppDock. It starts programs selected from the buttonbar, and (that's different) 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. To indicate the running program, the button goes grey. If you click that greyed button again, the application is closed. That's all. Almost. The top button provides some system options as about, move, setup and close. The second button provides an inboard 'Exit Windows' option.

Ini?

As most windows programs today **AppBar** has its own ini-file, called (you may guess now) **appbar.ini**. It looks like this:

```
[System]                          (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 appstart.grp       (any windows 3.1 groupfile,
startup.grp)
StayInFront=0                     ( 0 - 1, 0)
Initialized=2                     ( 0 - 2)

[Sounds]
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]
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 setup and change you appbar.ini file.

New?

In **AppBar** 3.30 the following things are new,
- *ParamsXX*= specify a optional parameter (DataFiles or StartUpParameters) given
  to a launched program.
- *SoundDirectory*= specifies the default directory for the sound file browser.
- AppBar supports multiple file Drag&Drop from the FileManager now. (It will launch a new
  instance of the buttonprogram for each file from the dropped filelist)
- added QuickLoad button to quickly setup AppBar with drag&drop and to copy buttonsettings to
  other buttons.
- added Run button, to launch program from a commandline.
- added Maximum Size option (ShowModeXX = 3), to start program as large as possible,
  with leaving AppBar visible.
- added dialogbox asking to save current position after moving AppBar.
- improved close function, closing program can ask for saving data first before terminating. (Thanks
  to *Erkki Riekkola* for providing crucial source).
- added new AppBar utility AppMore, which can launch different programs in different instances.
- increased maximum of buttons to 48.

- made AppBar restartable when it is Shell.
- AppBar will execute options given to win.com when it is Shell.
- fixed bug in maxcolumns setting.
- fixed bug in executing groupfiles: programs will be launched minimized now.
- Added minimize button. When AppBar is minimized just doubleclick on icon to restore AppBar.
- abandoned difficult Buttons/Columns settings. Now blank buttons are added to get enough buttons for the specified number of Columns.

Where?

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.

Shell?

Not only will **AppBar** work nice when you put it in your *load*= line in win.ini, but it also works great as shell. Check the checkbox "Use AppBar as Shell" in the system dialog window of AppSetup to install AppBar as shell. When AppBar is shell, it executes any given program from the win.com commandline, and the files specified by the *load*= and *run*= lines, and the applications in the groupfiles (seperated with a space) specified by the ***ShellGroup***= statement in appbar.ini. Make sure that the specified groupfiles are **created** in 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=&run= lines or from the ShellGroup anymore.

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, when it is an executable file, or the associatated 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 fron *.ico, *.exe, *.dll files to display icons on the buttonbar. This means that you can not only use the included iconfiles (appicons.zip) and iconlibraries (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 iconlibrary 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, 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 a 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 on (in QuickLoad mode), a special buttoncopy cursor will be visible. Just click on any other button (except the two systembuttons) to copy the settings from the first button. After the QuickLoad button is clicked again AppBar will return to its normal Launch mode.

Columns?

I have recieved a lot of mail about the use of the Buttons and Columns settings for AppBar. It wasn't always as clear as it should have been. I hope to make it clear now. With the Buttons statement you can specify how many buttons you want to have beside the two systembuttons with are always vivsble. The maximum of Buttons is 48, with 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 seperate column, so you need 18 vertical columns. Of course the maximum value of Columns is 50 (with is the maximum of Buttons plus the two systembuttons), 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.

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.

Windows?

This is version of **AppBar** is written for Windows 3.1 (with the standard MultiMedia extensions). It makes explicit use of mmsystem.dll, commdlg.dll and shell.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 aboutbox.
	Exit option built-in. Made use of Columns possible
	and increased maximum number of buttons up to 24.
3.1415a	Maintenance release, added possibility to specify a startup directory.
3.1415b	Maintenance, relase, added skipping closing feature, useful when using
	Windows 3.1.

3.14159

Upgraded to Win3.1. Changed redrawing code for icons. Added Setup program. Added Restart Option. Fixed bug in startDir function: current drive is also changed. Added click or DoubleClick mode. Added Start normal, maximized or minimized. Increased maximum buttons to 32. Load= and Run= statements in win.ini are supported when AppBar is shell. StayInFront option is added. Changed Aboutbox. Improved StayInfront. Skipped close option for all *.com and *.pif files. Start AppSetup when appbar not initialized. 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 dialog 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 minimized. 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 will it 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= & run= lines. If Buttons % Rows isn't zero, add Buttons up till it is. Added WM_APPBAR_RESTART message. Added minimize button. Added possiblity to copy buttonsettings in QuickLoad mode.

Ware?

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 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 catagory),
All other users,
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:
E. du Perronlaan 140, 2624 NA, Delft, Holland (You pay the first one, Okay?).

NEVER*never* *Software 1992.*