Half-Shell

Version 1.1

Half-Shell is a minimal replacement shell for Microsoft Windows 3.0's Program Manager. Its main design goal is to free up memory for other applications and, above all, make Windows load faster.

To use **Half-Shell**, you will need a PC-compatible system with an 80286 or higher processor, running Microsoft Windows 3.0 or later in Standard or 386-Enhanced mode. Use of a mouse is optional.

Quick Start

To install **Half-Shell** on your system:

1. If you received **Half-Shell** on a disk or unpacked it into a directory other than your main Windows directory (usually C:\WINDOWS), type **INSTALL** followed by the name of your main Windows directory. For example:

   INSTALL C:\WINDOWS

   INSTALL copys the files **HALFSHEL.EXE**, **HALFSHEL.INI**, **HALFCON.EXE**, and **SWAPSHEL.EXE** to your Windows directory. *If you have a previous version of* **Half-Shell** *installed on your system,* **INSTALL** *will replace the old version, without overwriting your launch list settings.*
2. Type **SWAPSHEL** at the DOS command prompt to make **Half-Shell** your current Windows shell.
3. Read Using Halfshell, below.

Exiting Windows

When using **Half-Shell** as your active shell, you can exit Windows by choosing **Close** from **Half-Shell**'s system menu. This should shut Windows down in a flash if there aren't any document changes to be saved. Windows will terminate only if all running applications agree to terminate.

Some Background

I wrote **Half-Shell** to meet one of the unfortunate demands of Windows programming -- that one swap into and out of Windows constantly while testing and revising an application. Program Manager -- mainly because it needs to perform a lot of group-management housekeeping -- loads slowly and (with Save Changes checked) exits even more slowly. Norton Desktop, much as I love it, is necessarily worse in this respect. And if your hard drive looks anything like mine, File Manager has a good-sized directory tree to read -- every time you load Windows...

This is where **Half-Shell** comes in. Because its EXE is much smaller than any of the others' (12K, compared with 54K for Program Manager, 104K for File Manager, and who knows how much for Norton Desktop, once all of its essential support files have been loaded) and it reads at most two INI files on startup, **Half-Shell** is much quicker to load and consumes much less memory.

Of course, such speed and compactness must come at the expense of some functionality. -- **Half-Shell** in no way attempts to duplicate the slick, icon-based file- and program-management facilities of its counterparts. However, it *does* provide basic facilities for running applications (which can even be one of the other shells), including a "launch list" that can remember up to 26 commonly-used commands for instant point-and-shoot execution. And the best part is, you don't even have to give up your current shell!

SwapShell

Provided with **Half-Shell** is a DOS command-line utility called **SwapShell**, which makes switching between **Half-Shell** and your favorite full-fledged shell a breeze! You run **SwapShell** the first time you want to install **Half-Shell** and each time you want to switch shells thereafter. **SwapShell** automatically modifies your Windows INI files to make the new shell active, and remembers the old shell setting so it can be swapped back in later.

To make **Half-Shell** load even faster when it is installed, **SwapShell** also takes the liberty of swapping your wallpaper setting (if any) with "(None)". You can prevent it from doing so by using the **-w-** command-line option when invoking **SwapShell**. Just like the shell setting, the wallpaper setting is remembered so it can be swapped back by a subsequent invocation of **SwapShell**.

To run **SwapShell**, simply type **SWAPSHEL** at the DOS command prompt. To suppress wallpaper-swapping, use **SWAPSHEL -w-**. Since it is installed in the Windows directory, **SWAPSHEL** can be run from anywhere.

Each time **SwapShell** alters your **WIN.INI** or **SYSTEM.INI** file, it first makes a backup copy of the file with a **BAK** extension (older **BAK**s are overwritten). **SwapShell** also creates a small file in your main Windows directory called **SWAPSHEL.DAT**, in which it records the currently swapped-out shell and wallpaper settings.

Using Half-Shell

Once **SwapShell** has installed it, **Half-Shell** loads automatically whenever you run Windows. It appears initially as a title and menu bar at the top of the screen, though you can use its minimize box to shrink it down to an icon, and still access all of its features through the icon's system menu (click once on the icon to bring it up). If you prefer to have **Half-Shell** always run as an icon, you can specify this in **Half-Shell**'s configuration. (See Customizing Half-Shell.)

Foremost in **Half-Shell**'s menu bar is the **Launch** menu, which allows you to choose from up to 26 predefined commands to execute. (A command can simply name an application's EXE, or even include command-line options and names of documents to load, etc.) You will probably want to customize this menu to include applications that you use often (and remove defaults that simply get in the way); it initially contains entries for all the standard Windows accessories. (The next section tells you how you can customize the **Launch** menu for your own use.) Choosing an item from this menu tells **Half-Shell** to execute the associated command-line.

To supplement the **Launch** menu, there is the **Run** command, which brings up a dialog box that allows you to type in the name of an appliction to run, along with any desired command-line options. This provides a way to run less-frequently-used applications that aren't on your **Launch** menu. By checking the "Run minimized" box, you can also specify that the application is to be run as an icon (though it is up to the application whether it will respond to this "polite suggestion").

The **Configure** command brings up a dialog box that allows you to tailor **Half-Shell**'s operation to suit your own needs (see Customizing Half-Shell, below).

Together, **Launch** and **Run** comprise the sum of **Half-Shell**'s functionality. Clearly, there isn't much to it, but then that's exactly the point -- essential capabilities, minimal system burden.

Customizing Half-Shell

Once you have installed and experimented with **Half-Shell**, you will probably want to customize it to better assist you in the way you work. **Half-Shell 1.1**'s new configuration program makes this easy to do, even on-the-fly.

Selecting **Configure** from **Half-Shell**'s main menu brings up the "Half-Shell 1.1 Configuration" dialog, which allows you to modify the **Launch** list and set some options that change the way **Half-Shell** operates. Using the configuration dialog is pretty straightforward:

Add, **Edit**, and **Delete** provide facilities for modifying the **Launch** list; **Edit** and **Delete** operate on the currently selected launch item in the adjacent list box. A launch-list item has three attributes: a quick-selection key, a title (which appears in the **Launch** menu), and the command line that is executed when the item is selected. The command line may be up to 255 characters long, and is the name of an application's **EXE** (preceded by a path to it, if necessary), optionally followed by one or more command-line options, filenames, etc. to be passed to the application. Some examples of possible command lines:

```
reversi
write.exe
c:\windows\games\tetris.exe
write grtnovel.wri
```

The three check boxes at the top of the dialog specify whether **Half-Shell** is to be run as an icon or a top-of-screen menu bar, and whether it should execute the applications named in **WIN.INI**'s "load=" and "run=" settings as a part of the Windows startup process.

Incidentally, **Half-Shell**'s configuration facility is its own, separate program. -- Therefore, though it's always immediately accessible via **Half-Shell**'s **Configure** menu item, it doesn't slow the Windows startup process, or take up precious memory when it isn't in use.

Manually Editing HALFSHEL.INI

Using the **Configure** dialog is the preferred method of customizing **Half-Shell** beginning with version 1.1. However, for backwards compatibility (or those who care to snoop), here are the revised instructions for manually modifying **HALFSHEL.INI**:

When **Half-Shell** is run, it loads the **Launch** menu items from **HALFSHEL.INI**, which is located in the main Windows directory; by editing this file (using Windows Notepad or another ASCII text editor), you can change the default settings to anything you like.

Each item in the **Launch** menu corresponds to a letter in the alphabet (which will be its quick-selection character), and is defined by two variables in **HALFSHEL.INI** -- one specifies the title that will be displayed in the menu, and the other specifies the program to be executed along with any desired options and filename parameters. As an example, to set item G to bring up your masterpice work-of-fiction-in-progress, you might use:

```
TitleG=Bob's Great American Novel
CommandG=WRITE.EXE c:\bob\thebook.wri
```

Some additional examples:

```
TitleN=Norton Desktop
CommandN=NDW.EXE
```

```
TitleD=Turbo Debugger for Windows
CommandD=C:\BORLANDC\BIN\TDW.EXE

TitleW=Whitewater Resource Toolkit
CommandW=C:\BORLANDC\BIN\WRT.EXE
```

Titles may be up to 70 characters long, commands up to 255 characters; up to 26 launch items can be specified. If you edit **HALFSHEL.INI** from within Windows while **Half-Shell** is active, you'll need to exit and re-load Windows for the changes to take effect.

An additional variable in **HALFSHEL.INI** specifies whether **Half-Shell** is to be run as an icon or as a bar across the top of the screen. If you set **IconizeShell=1**, **Half-Shell** will initially appear as an icon; if you set it to **0**, **Half-Shell** will appear as a title and menu bar across the top of your screen.

Finally, there are the **LoadLoadItems** and **RunRunItems** settings, which tell **Half-Shell** whether to execute the applications named in the "load=" and "run=" settings in **WIN.INI**, respectively. Like **IconizeShell**, these variables can be set to either **1** (for "yes") or **0** (for "no).

Eeek! A Bug!

Well, let's hope not... I've put **Half-Shell** through every test I could think of, and in fact tend to use it more as my regular Windows shell now than as a substitute. *But* ... if when you're kicking all the tires something goes wrong or appears to, however minor, I'd definitely appreciate hearing about it, so we can see about making future releases even more invincible.

I'd also love to hear your ideas and suggestions for enhancements to **Half-Shell**. What would you like to see added, changed, or even removed? And please bear in mind that **Half-Shell** is built to be lean-'n-mean, not all-powerful. The more features, the more program code, and the larger and slower **Half-Shell** becomes. In designing this first release, I tried to make the best all-around efficiency-vs-convenience trade-offs that I could.

The best ideas for additions would involve a minimal amount of additional code.

You can reach me by U.S. mail or e-mail; the quickest way is probably via the SmartNet "C" conference. Otherwise:

Troy Stephens
2315 Overland Ave.
Los Angeles, CA 90064-2209

CompuServe ID: 76625,2402

License and Copyright

Half-Shell 1.1

There is no registration fee for Half-Shell 1.1 per se, though a modest contribution of $5 or so would make a hard-working college student's day (!) and help to insure the future of Half-Shell as an evolving product. Thanks in advance to those who can cointribute!
-- And *Enjoy!!!*

Trademarks