

STARTUP

Version 1.0

The Fast Way to Start Windows™ (and other) Applications

Copyright © Jeff Lansing 1993
All Rights Reserved

Notice

STARTUP is so simple as to be **almost** foolproof; nevertheless, STARTUP is distributed "as is" with no expressed or implied warranty of any kind.

IN NO EVENT SHALL JEFF LANSING BE LIABLE OR RESPONSIBLE FOR ANY LOSS, DAMAGE, INCONVENIENCE, OR OTHER PROBLEM CAUSED BY USING STARTUP. THE ENTIRE RISK OF USING STARTUP IS YOURS.

1. Intro

Welcome to STARTUP, the program for starting all the Windows™ (and/or non-windows) applications you could ever want with a single (double-)click of your mouse, or a single keystroke.

STARTUP does not require any programming skills, it does not require learning any new commands, and it does not mess with your computer system. In other words, STARTUP is user friendly.

As an enthusiastic user of Windows™, I find myself doing almost everything with them. But the sort of work I do requires having several windows open at once, each one requiring its own "document file" and parameters, and I cringe a little every time I have to get all that stuff going. Now I happen to know, or at least I think I know, just about all the ways there are (including some that are still in Beta testing) of starting Windows™ application programs, but very few of them seem to be quite flexible enough to do all that I want to do (and those few that do require taking the time to learn a special language). On the other hand, by using STARTUP I find that I can easily save myself time, and I can avoid that little bit of stress that accompanies useless repetitive activity.

2. Registration

2.1 How To Register

STARTUP is shareware. Users can use an unregistered copy of STARTUP for 21 days. After that time users must either register their version of STARTUP or delete STARTUP from all storage media.

To register send \$10.00 to the following address:

Jeff Lansing
4354 Witherby Street
San Diego, CA 92103-1150

Registered users are entitled to technical support via phone, and to substantial discounts on more advanced versions, as they appear.

2.2 Bug Reports/Enhancement Requests

Please send all bug reports and enhancement requests to the above address, or to:

lansing@cogsci.ucsd.edu or lansing@bend.ucsd.edu

2.3 Distribution

Since STARTUP is shareware you are encouraged to distribute it freely, as long as all the files in the original are distributed together.

3. Getting Started

3.1 How STARTUP Works

STARTUP takes a single argument — a "document file", in Windows™ terminology — which may either be passed to STARTUP as described below, or may be the default command file **startup.lst** ('**lst**' means "**list** of commands", but of course you can use any extension you want for your command files) which is residing in the current directory.

Note: if no argument is passed to STARTUP, and there is no file with the name **startup.lst** in the current directory, the advisory message "Startup can't open file: startup.lst" will appear.

STARTUP regards the lines of the command file which is passed to it — except for lines which begin with a # sign, which is there to allow you to make notes to yourself within command files — as a series of commands which are to be executed (think of the commands in a DOS batch file, if you like), and it proceeds to carry out those commands. If it comes to a line (i.e., a command) which for some reason it cannot find, the advisory message "Startup can't start:" followed by the troublesome command appears. When you respond to this message STARTUP will then continue carrying out any following commands.

Note: Nothing prevents you from having any (or even all) of the commands in your command file be **startup.exe** commands. I personally regard this as a powerful feature which allows me to "chain together" command files which I have used previously, just as they are. However, **TAKE CARE** not to include a command file which is near the head of the "chain" into another command file which is near the tail of the same "chain", as is done in the following example (which is somewhat extreme, because in fact the same file includes itself):

```
#command file named example.lst
write startup.wri
startup example.lst
```

Doing so will usually result in all of the Windows™ resources getting used up, which will slow Windows™ to a halt. (There are a few benign cases in which STARTUP can catch itself doing this, and stop, but **DON'T** count on that happening.)

Finally, one of the more useful features of STARTUP is that it understands about drives and directories, so that you don't have to worry about them. The central principle here is that **a command file refers to programs which are on your PATH and to documents which are in the same directory as it is in**, unless you specifically tell it that you want some other drive or directory. What this means to me is that I save time, and I don't have to concern myself with whether all that truly *mysterious* directory and drive stuff is going to work out alright, or not.

3.2 Preparing command files for STARTUP

Any ASCII text editor may be used to prepare command files for STARTUP. For preparing the simple files that STARTUP uses, NOTEPAD is quite good enough. Start NOTEPAD (by double-clicking its icon in Program Manager, say), enter the command lines and any comment (i.e., "#") lines you want, and Save this command file in the directory where the documents which it refers to happen to reside. It's that simple.

3.3 Running STARTUP

There are several different ways to run STARTUP itself once its command file(s) are ready. Here are a few suggestions (number 1b is my current favorite):

1) Using File Manager, you can:

- (a) Select File|Run from the menu, type `startup startup.lst` (for example), and press the enter key (or button).
- (b) Select File|Associate from the menu, and associate files with the extension `startup.exe`. Then you can double-click on any file with that extension.
- (c) Move the cursor over the file `startup.lst`, hold the left mouse button down, drag it to the program `startup.exe`, and release the left mouse button.
- (d) Double-click the `startup.exe` icon. (The file `startup.lst` must be in the same directory.)

2) Using Program Manager, you can:

- (a) Drag files with the extension `lst` out of File Manager and drop them into a Program Manager group. (This assumes that you have associated them, as in 1b above.) A



icon with the name of that command file (but without the `lst` extension) will appear in that Program Manager group window, which you can then double-click on.

- (b) Select File|New from the menu, choose New Program Item, and enter: `startup c:\mydocs\othrdocs\startup.lst` (for example) as the Command. The effect will be the same as in 2a above.

3) Using Norton Desktop © (which some people like) you can:

Drag `.lst` files out of a Drive and drop them on the Desktop. The resulting icon buttons work just like BatchRunner buttons (but without having required you to learn that **arcane** BatchBuilder language).

3.4 Working with Drives and Directories

As mentioned, the central principle for working with drives and directories is that **a command file refers to programs which are on your path and to documents which are in the same directory as the command file is**, unless you specifically tell it that you want some other drive or directory. What this means is that you only have to type the name of the program and the name of the document into the command file.

But what if you want to start one group of tasks in one directory, and another group of tasks in another directory, both at the same time, without having to specify drives and directories for every task in the

group?

In this case you just make a **startup.lst** file for the files in the one directory, and save it in that directory, and also make a **startup.lst** file for the files in the other directory, and save it in its directory, and then you simply add a command like the following: whatever is appropriate) to one of the command files. (Here is where you specifically tell STARTUP to change directories.) And of course this method is not limited to just two groups of tasks, in two directories, but can be used for as many as you choose.

Note: Suppose you (quite cleverly) figured out the following time-saving technique: "Sometimes I'm working in one directory, and sometimes in another. So why should I have to (sometimes) change back to the other directory in order to start both groups of tasks? Why don't I just add a command like: `startup c:\mydocs\othrdocs\startup.lst` (or whatever) to BOTH of the **startup.lst** files so that, no matter which directory I am working in, the tasks in both directories will get started up?" Well, the answer is that this would turn out to be one of those endless "chains" described in a note above, and would not be quite what you wanted.

Note: A simple solution to the above problem is to add another command file to EACH directory:

```
#command file named both.lst in directory c:\mydocs\othrdocs
startup startup.lst
startup d:\yourdocs\thisdocs\startup.lst

#command file named both.lst in directory d:\yourdocs\thisdocs
startup startup.lst
startup c:\mydocs\othrdocs\startup.lst
```

In this way you can start both groups of tasks (listed in the respective **startup.lst** files) by double-clicking EITHER **both.lst** file, as desired. And this gives you a structure of STARTUP command files which you can later build on.

3.5 Hints and Tips

There are many DOS command that can be run with STARTUP, such as `dir` and `find`. To keep the results of these commands on your screen long enough for you to read them, however, you need to use Pifedit (which is usually in your Program Manager "Main" group) to change the **_default.pif** so that the Close Window on Exit box is not checked. In this way you can include commands like the following in your STARTUP command files:

```
find /n /i "startup" readme.txt
```

Other DOS commands (those which don't have a separate `.exe` or `.com` file in the `\dos` directory) require you to start the **command.com** interpreter in order to run them, as in the following example, which could be a line in a STARTUP command file:

```
command.com /c for %g in (*.*) do find /n "startup." %g
```

Note: The DOS functions `|` ("pipe") and `<` ("redirection") also require this, as in the following:

```
command.com /c find /n /i "command file" startup.wri | more
```

Note: You can substitute **dosprmt.pif** for **command.com** in the above, in which case you will

want to change **dosprmt.pif** instead of **_default.pif** to keep the window from closing too soon.