SmilerShell Help

To find out more about SmilerShell, choose a topic below:

Introduction

Windows makes many things easier, but it also makes some things harder. Even in this era of the graphical interface, there are tasks that can be done much more easily by typing in a command than by menus and pointing and clicking and such. That's what SmilerShell is for.

SmilerShell is the ultimate Windows command line. It runs anything (DOS programs, Windows programs, or DOS internal commands), and supports pipes and redirection. It has a command line editor with history and search, aliases (type-in or on the function keys), a fast-directory-change utility that works across multiple drives, a place to list your favorite applications (click on one and it runs), and many helpful Windows functions like a clock in the title bar and a real-time system resources report in the menu bar. After it pops up it's still a very compact window, but to make it even smaller you can toggle away the menu or title bar.

The built-in command line editor saves all submitted commands, allowing you to get back a previously-submitted command, change it, and re-submit it. You can have SmilerShell search for a previous command; no need to scan them all yourself to find the one you want. You can load a command stack from a file when you start, or manually at any time. You can save the current command stack to a file, to load later or edit as needed.

SmilerShell's aliases are short commands that are replaced with longer commands. Aliases can be like regular commands, just type them in. Or they can be on function keys, hit the F-key and it runs, no need to press Enter.

SmilerShell's fast directory-change utility is called DC. Just type DC and the first few characters of the directory you want to be in, and SmilerShell takes you there. If your command is ambiguous, a window pops up, letting you choose which directory you want. This works across as many multiple drives as you tell it to be aware of.

SmilerShell can stay on top of all other windows, show the current directory and/or a clock in its title bar, display Windows free memory and resources in its menu bar, and directly manipulate the inactive windows that remain when you run DOS commands from Windows. SmilerShell takes up very little space on your screen, but to save more space you can even remove the menu and/or title bar entirely.

Related Topics:
Quick Start / Hints And Tricks
Why Is This A Shell?
The Initialization File
Menu Items
Submitting Commands
Using Arrows To Retrieve Previous Commands
Aliases

DC: Directory Change The Fast Way
Command Stack Files

Shareware: Try Before You Buy

Thank you for trying SmilerShell. You are welcome to test it for a week or two. I hope you like it.

SmilerShell is marketed as shareware. Try it on your own system to see if it meets your needs. If you find it useful and you keep it on your system for more than two weeks, you are obligated to send in the registration fee. If you don't find it useful, simply delete it from your system.

Or for those who want the very best, consider SmilerShell Pro, which does everything that regular SmilerShell does, and in addition adds some rather nifty features not found in the regular SmilerShell. To find out more about SmilerShell Pro, click here:

To order, send $19.95 for SmilerShell or $29.95 for SmilerShell Pro (plus $3.50 for shipping) to:

Bardon Data Systems
1023 Key Route Blvd.
Albany, CA 94706

Outside North America please add $6 for overseas shipping charges.

You can order SmilerShell or SmilerShell Pro through Bardon, or through our toll-free telephone order-taking service (800) 242-4775 (weekdays 7 to 6 Central time), or on Compuserve, or through distributors worldwide. Details and addresses are in the file REGISTER.TXT.

If ordering through Bardon, you can print and mail invoice.wri, the invoice that came with this package. It's in Windows Write format. With a MasterCard or Visa you can order by phone, at (510) 526-8470, in which case you'll be given your registration number immediately so you can get rid of those reminder screens right away. Or simply mail in your card number and expiration date.

Registered users get a registration number that will turn off the reminder screens, a printed manual, support, update notices, and a disk with the most recent version.

Registered users also get these Extra Free Bonuses: Two more handy Bardon utilities (WHATSNEW lists files not yet backed up, or directories containing such files; PR/PRFILTER format output for printing, and add a header with filename, size, create date, and print date), discount certificate for JCSM shareware CD-ROMs at half price, discount on PsL shareware-by-mail (up to 2/3 off!), free Compuserve startup kit, other Windows shareware I think you'll like, and whatever other goodies I can fit on the disk.

All these bonuses are also included with SmilerShell Pro.

SmilerShell is produced by a member of the Association of Shareware Professionals (ASP), which supports quality shareware development through a number of mechanisms. For more information, see the ASP STATEMENT elsewhere in this documentation.

Related Topics:
ASP Statement

What Happens When You Start SmilerShell

When you start SmilerShell, it first reads the initialization file. By default this is the file "smishell.ini" in the same directory as the SmilerShell program. However, you can specify a name and directory for this on the SmilerShell command line. Values are set, based on your entries in this file. If you start SmilerShell without an initialization file, you'll be asked if you'd like SmilerShell to create one and fill it with reasonable values, then let you edit it in Notepad before proceeding.

Next, SmilerShell looks for DC information. It will rescan your directory structure if you've set the ini file parameter startupDCscan toTRUE. If not, SmilerShell looks for the file it creates when you Scan Directories. By default it is named "smishell.dir" and is in the same directory as the initialization file. You can specify a different name and directory for it by using the dirfile= ini file parameter.

Finally, if you have set winwidth to PREV in the initialization file, SmilerShell sets itself up the way you left it last time, and changes to the directory you were in when you last used it.

Related Topics:
The Initialization File
DC: Directory Change The Fast Way

Menu Items

SmilerShell has five major menu items: File, Edit, Options, Apps, and Help. The unregistered version has a Register menu item, which provides information on how to register SmilerShell. Using the Options menu, you can also toggle another item onto the menu bar: a real-time report of available Windows memory and resources.

Related Topics:
The File Menu
The Edit Menu
The Options Menu
The Apps Menu
The Help Menu
The Register Menu Item
The System Resources Menu Item
Keyboard Shortcuts

The File Menu

The File menu starts with the traditional items New, Open, Save, and Save As. These items let you manipulate your current command stack (list of stored commands). Next on the File menu are the items DOS Box, Run, List Aliases, List Commands, Scan Directories, Edit Ini File, and Exit.

New clears the command stack. That is, it makes your list of previously issued commands go away.

Open lets you choose a command-stack file and read it in. It clears the current command stack, and reads in a new command stack from the file. If you tell it to read an ini file (i.e., give a filename with the extension "ini"), SmilerShell assumes the ini file has a section called [SmilerShell Params] with an entry called cmdstack= that contains commands to read in. If you tell it to read a file with any extension other than "ini", SmilerShell assumes that it's an ASCII text command-stack file, with one command on each line.

Save saves your current command stack to a file using the most recently set command-stack or ini file name. If it's an ini file, the entry is saved in a section called [SmilerShell Params] in an entry called cmdstack=. If not, the commands are saved in a plain text ASCII file, one command per line. There is a 300-character limit on ini file stacks.

You set a command-stack file name (to be used by the Save item) by specifying one in the Open or Save As dialogs during the current session, or by setting the cmdfile= parameter in your ini file. If you haven't explicitly set a name yet, the name "smishell.stk" is used. By default it's assumed to be in the same directory as the ini file.

Save As asks for a filename, then saves the current command stack to that file. If it's an ini file, the entry is saved in a section called [SmilerShell Params] in an entry called cmdstack=. If not, the commands are saved in a plain text ASCII file, one command per line. There is a 300-character limit on ini file stacks.

DOS Box gives you a DOS session, full screen or windowed depending on how you've set the Options menu item
DOS In Window. Type Exit at the DOS prompt to return to SmilerShell.

Run lets you choose a program from a file/directory dialog box. The filename you choose is placed on SmilerShell's command line, where you can add any needed parameters before submitting it.

List Aliases shows you what command substitutions are currently in effect. It shows both your command-line aliases and your function-key aliases. If you don't like the way they look, there's an Edit Ini File pushbutton. You can run an aliased command

from here by clicking on it, or fetch it into the main window for editing.

List Commands shows a list of every command you ran from the command line during this session, plus any commands preloaded at startup. You can submit a command from here by clicking on it, or fetch it into the main window for editing. There's also an Edit Ini File pushbutton.

Scan Directories generates an internal list of all directories on each drive listed by the scandrives= parameter in your ini file (default is to just scan drive C). It saves this internal list to the DC info file, either a filename you specify or the default "smishell.dir" in the SmilerShell ini file's directory. DC's "/r" parameter can also generate this list.

Edit Ini File sets you up to edit your initialization file in Notepad.

Finally, the File menu has an Exit item which terminates SmilerShell.

The Edit Menu

The Edit menu starts with the standard Windows features Undo, Cut, Copy, and Paste for sending information to and from the Windows clipboard, and Clear to delete selected text in the input window.

There is also a Remove Inactives item, which searches out and closes all the "(Inactive..." windows on your desktop. If you've toggled the Options menu item Inactives Stay Visible to display inactive DOS windows after their command terminates, you'll find that these windows accumulate quite rapidly. Remove Inactives (or its keyboard equivalent Alt+R) makes them all go away.

Related Topics:
The Options Menu

The Options Menu

The Options menu items toggle on and off various SmilerShell features.

Clock (Alt+C) puts an hour:minute clock in the title bar.

System Resources (Alt+S) shows available system resources in the menu bar: memory, GDI, and User resources.

Directory (Alt+D) show the current directory in the SmilerShell title bar.

Overtype (Alt+O) toggles the command line between insert mode and overtype mode. When it's in overtype mode a flag appears in the title bar, over the second 'e' in 'SmilerShell'.

Topmost (Alt+T) makes SmilerShell a topmost window, so even when inactive, it sits on top of other windows.

Title Bar (Alt+L) hides the title bar. This saves screen space. To move SmilerShell on the screen without a title bar, click the right mouse button in the edit area and hold it down while moving the window. Note: toggling the title bar consumes User resources.

Menu (Alt+M) hides the menu bar, making SmilerShell even smaller. When the menu bar is hidden, a "Show SmilerShell Menu" item is added to the System menu. Or use Alt+M to get it back. The other keyboard accelerators (Alt+C, Alt+D, etc.) also continue to work properly when the menu is hidden.

Inactives Stay Visible (Alt+I) controls whether, after SmilerShell runs a DOS command, the command's inactive window sticks around or goes away. Keeping those inactive windows around can be quite handy, letting you see the results of previous commands, but they do eventually clutter your screen. True, you can make them all go away using the Remove Inactives item on the Edit menu (or simply type Alt+R). But if you don't want to see them in the first place, you can simply toggle Inactives Stay Visible off. Or to run one command as if Inactives Stay Visible is set to the opposite of its current value, start it with an asterisk (for example *dir).

DOS In Window (Alt+W) controls whether SmilerShell's DOS commands run fullscreen or in a window. Or to run one command as if DOS In Window is set to the opposite of its current value, start it with a right-bracket (for example >copy foo.bat b:).

Apps Menu controls whether you show or hide the Apps top-level menubar item. Hiding the Apps menubar item saves menubar space, which is sometimes handy.

Related Topics:
Submitting Commands
DOS Commands: Fullscreen or Windowed

The Apps Menu

The Help Menu

The Help menu has two items: Help and About SmilerShell.

Use the Help item (or press F1) to get on-line help about SmilerShell.

The other item, About SmilerShell, is a typical About box. It shows the SmilerShell version number, contact information, and your registration number, if you are a registered user.

The System Resources Menu Item

When you toggle this on, using the Options menu item System Resources, the System Resources Menu Item provides a report on three key Windows resources: bytes of available memory, GDI resources, and User resources. It changes in real time to show your currently available resources. Although it is on the menu bar, it has no menu associated with it.

You may wonder why the System Resources menu item doesn't display System resources, as displayed in Program Manager's About box and other places. It turns out that "System resources" is just Windows shorthand for "the smaller of User and GDI resources." Why take up screen space with information you've already got?

Related Topics:
The Options Menu

Submitting Commands

When SmilerShell has the input focus, simply type any command, just as you would at the DOS prompt. You can run Windows programs, DOS programs, or DOS internal commands like DIR or TYPE. You can use CD or CHDIR to change SmilerShell's current directory. (Or you can change directory a lot faster with the built-in SmilerShell command DC.) You can use the built-in SmilerShell command SHOW to display, and then select, filenames.

Because SmilerShell supports File Associations, quite often you only have to give the filename, without naming the program to run it. SmilerShell can often tell what program to run, just by looking at the filename. For example give "FILENAME.WRI" and SmilerShell knows to run Windows Write on this file. This works very nicely with SHOW: use SHOW to pick a file, then just press Enter to run the proper program with that file.

DOS commands called from SmilerShell will run fullscreen or in a window, depending on how you have set the DOS In Window menu item.

Or you can start any command with a right-bracket to toggle DOS In Window for just that one command (for example >copy foo.bat b:).

If you have toggled Inactives Stay Visible to allow it, then after SmilerShell runs a submitted DOS command, the final results are displayed in an inactive window. That is, a window titled "(Inactive SOMETHING)".

When an "(Inactive..." window gets the focus as a result of running a command, SmilerShell actively takes the focus back again, so you can continue running commands from SmilerShell. Because Windows requires it, SmilerShell pauses briefly before attempting to take back the focus. By default this pause is 1000 milliseconds (1 second), but you can set it by using the ini file parameter restoretime=. How fast can you get away with on your system?

You can use the Edit menu option Remove Inactives (or simply press Alt+R) to destroy all the "(Inactive..." windows on your desktop.

Or you can start a command with an asterisk (for example *copy foo.bat b:) to toggle Inactives Stay Visible for just this one command.

Every command you submit is tested to see if it matches anything on the list of aliases you provided in SmilerShell's initialization file. If the first word of your command matches an alias, the replacement for that alias is substituted for the first word of the command you typed.

Or to skip alias testing for this one command, start it with a equals sign (for example =list). The command line will be run just as you typed it.

An easy way to remember what each of the three command line flags does is:

- * Flash! It's gone! (or, Flash! A window appears!)
- > Moves from larger to smaller, or smaller to larger
- = Does just what it says (no alias substitution)

The three flags (* > =) can be in any order. For example, ">*=dir *.exe" could be run.

When SmilerShell finds a PIF for any program, it uses the PIF's Inactives Stay Visible and DOS In Window settings instead of its own, assuming that if you set up a PIF you did so for a reason.

By default, SmilerShell uses the command processor listed in the COMSPEC environment variable. This is usually DOS's COMMAND.COM, but some people use alternate processors such as 4DOS or NDOS. Sometimes the alternate processor has its own PIF file, for example 4DOS.COM may have a 4DOS.PIF file. This PIF will prevent SmilerShell from controlling the Inactives Stay Visible and DOS In Window settings itself, since SmilerShell will defer to the settings given in the PIF. To change this, either delete the PIF or set UseComspec to FALSE so SmilerShell will explicitly use COMMAND.COM, which it can control via _default.pif.

Related Topics:
Using Arrows To Retrieve Previous Commands
Editing Commands
DOS Commands: Fullscreen Or Windowed
The Initialization File
DC: Directory Change The Fast Way
SHOW: An Alternative To DIR

Using Arrows To Retrieve Previous Commands

When you submit a command by pressing Enter, SmilerShell stores it internally in a command stack. To retrieve a command, press the up/down arrow keys until the command you seek is displayed. Press the up-arrow to see the previous command, or the down-arrow to see the next command.

You can search for a particular previous command to be displayed. Let's say you want to find the command "dir \windows\system*.ini /p" that you ran some time before. Just type D before you press the up-arrow. SmilerShell will find the most recent command that started with D. You are not limited to just the first letter; you can type as much of the previous command as you need to specify the match you want. If the first match isn't the command you are looking for, press that arrow key again until the command you want comes up. The same match-string is used until you type something that changes a displayed command. Matches are not case-sensitive. The command-line flags (* > =) are ignored when checking for a match.

To simply retrieve all commands in order, just make sure the command line is blank when you first press the arrow key. You can clear the command line by pressing Escape.

You can also use the File menu's List Commands item to gain access to your entire command history.

Related Topics:
Submitting Commands
Editing Commands

Editing Commands

The normal editing keys allow you to move within the command line. Use Home, End, left-arrow and right-arrow to move within the command line. Ctrl+left-arrow move one word to the left, and Ctrl+right-arrow move one word to the right. You can clear the command line by pressing Escape.

SmilerShell's command line can be in either insert mode or overtype mode. Toggle this with the Options menu item Overtype, or just type Alt+O. In overtype mode, a flag appears in the title bar.

Related Topics:
Using Arrows To Retrieve Previous Commands
Submitting Commands
The Options Menu

Size Of Window

SmilerShell will accept commands of up to 128 characters (the DOS command line limit). You can make the command line window as wide as you like. However, there is never any need to make it more than one line high! If you try, it snaps back.

When maximized, SmilerShell takes up only the top line of your screen. You can set up a very useful configuration by setting SmilerShell as a topmost window, then maximizing it.

For a smaller SmilerShell window, use the Options menu to toggle off the menu and title bar, then mouse the window as small as you like.

Getting Rid Of Inactive Windows

The Options menu item Inactives Stay Visible controls whether, after SmilerShell runs a DOS command, the command's inactive window sticks around or goes away. Use the initialization file parameter showinactives= to set this to your preference at startup. If you don't specify a preference, then at startup SmilerShell sets the menu toggle to match the current systemwide value found in the "_default.pif" file in your Windows directory.

If you've toggled Inactives Stay Visible to allow it, each DOS command ends by firing up its own "(Inactive..." window. This is handy, letting you see the results of previous commands, but it does eventually clutter your screen. To make them all go away, use the Remove Inactives item on the Edit menu, or simply type Alt+R.

To toggle Inactives Stay Visible for just one command, start the command line with an asterisk (*list foo.txt for example). You can use this with the equals sign or right-bracket flags if you want to (for example =*>dir).

Related Topics:
The Edit Menu

About Internal DOS Commands

SmilerShell runs most DOS commands in a subshell. For internal DOS commands that affect the working environment, this is tricky. The subshell starts up with a copy of the parent's environment, things like current directory, environment variables, settable DOS version, etc. If you alter an environment variable or change directory in a subshell, the parent shell's information does not change. SmilerShell can support all the internal DOS commands you're likely to want. In addition, there are four "semi-supported" internal DOS commands: CHCP, PATH, SET, and VER. In DOS, these can both set and show environment values. If you enter one of these, SmilerShell will show their current value. However, because you are in a subshell, not your actual environment, you can't change these values in your actual working area through SmilerShell (or through Windows generally).

To summarize:

Supported Internal DOS Commands: CD, CHDIR, COPY, DATE, DEL, DIR, ERASE, MD, MKDIR, REN, RENAME, RD, RMDIR, TIME, TYPE, VOL

Semi-supported Internal DOS Commands: CHCP, PATH, SET, VER

Unsupported Internal DOS Commands: CLS, CTTY, EXIT, PROMPT, VERIFY and the batch file commands.

Aliases

An alias is a short command that is replaced with a longer command. Some people call them macros. There are two kinds of aliases in SmilerShell. In the first kind of alias, you type a (typically, short) command line and press Enter, and the first word of the line is replaced with another (typically, long) string. The rest of the original command line is tacked on after the replacement string. You can define up to about 100 of these type-in aliases. In the second kind, you press a function key and a predefined command is submitted. You can define one of these function-key aliases for each of F2 through F12 (F1 is reserved for Help).

Type-In-Aliases: Let's look at the first kind. Say you set up the alias:

```
dirprog=dir c:\develop\source\*.*
```

Whenever you enter the command "dirprog", SmilerShell will replace it with, and actually submit, the command "dir c:\develop\source*.*" to be run. This saves wear and tear on your typing fingers.

You can put parameters on this kind of alias. In our example, you could enter

```
dirprog /o /p
```

and SmilerShell would run the command

```
dir c:\develop\source\*.* /o /p
```

by adding the original parameters after the substituted alias.

A typed alias is used just like any other command; type it in (with parameters if any) and press Enter.

Unless you tell it otherwise, SmilerShell looks at the first word on each command line to see if it's an alias. To avoid alias checking for a particular command, start it with an equals sign. For example, if you actually had a program called "dirprog" that you wanted to run instead of the alias defined above, you could submit this:

```
=dirprog
```

Because the command line starts with an equals sign, SmilerShell skips the alias testing for this command.

You can use the equals sign flag with the right-bracket or asterisk flags if you want to. In this example, you could type *>=dirprog and press Enter.

Function Key Aliases: The second kind of alias is where you attach a command to a

function key. Just press the function key and the command is submitted. You don't need to press Enter to submit it. Function keys F2 through F12 can be set up this way (F1 is reserved for Help).

For example, let's say you have set up the alias:

```
(F5)=copy c:\develop\source\*.* b:\
```

Now, whenever you press F5 in SmilerShell, the command "copy c:\develop\source*.* b:\" will be submitted. It's very handy, no need to press Enter.

In General: Aliases are defined in the SmilerShell initialization file, in the section [SmilerShell Aliases], one per line, in the form alias=replacement. The left side of a function-key alias definition has the key name in parentheses, as in the example above. The left side of a type-in aliases can be whatever you like, as long as the alias-part has no embedded spaces.

By default, alias testing is case-sensitive. You can change this with the initialization file parameter AliasesCaseSens=.

Aliases can reference other aliases, but be careful to avoid self-referencing infinite loops (alias 1 defined in terms of alias 2 which is defined in terms of alias 1 which is ...).

Related Topics:
The Initialization File

DC: Directory Change The Fast Way

DC (Directory Change) is a built-in alias that lets you change directory very quickly. Instead of having to type in the entire pathname, you only need to give it the first few letters of the endpoint (leaf-node) directory you want. For example, instead of typing "cd \c8\mfc\samples\fileview" you could type "dc fi" and press Enter. If "fi" is enough to unambiguously specify one directory, DC takes you right there. If what you typed is ambiguous (maybe there's more than one directory whose name starts with "fi") a window pops up, showing all your possible matches in alphabetical order. The first possible match is highlighted. If there was no possible match, nothing is highlighted. Double-click on your choice, or single-click and press OK. At the top of the box is the number of directories DC knows about. There's a button to re-scan the directory list as well.

To re-scan the list and change directory all at once, use the /r parameter. For example "dc /r fi" would first re-scan, and only then look for that "fi" directory (in the new list) as described above. You can use a dash ("-r" instead of "/r") if you prefer.

If the endpoint directory is on a different drive, DC will first change drives, then change to the desired directory. There's no need for you to manually change drives first. DC does it for you.

By default, the DC data is stored in the file "smishell.dir" in the same directory as the initialization file (you can specify a name and location for this file with the ini file parameter dirfile=). SmilerShell creates this file the first time you use DC, or whenever you use the File menu item Scan Directories. It contains the name of every directory on each drive that was scanned. These are the directories that DC can change to. To indicate what drives you want scanned, set the ini parameter scandrives=. For example, if this is in your ini file:

```
scandrives=cdm
```

then SmilerShell will generate a list of all directories on your c, d, and m drives. (The list of drives can be separated by commas or spaces [scandrives = c, d, m], or bunched together as in the example above.)

If the directory layout of your system is constantly in flux, you may want to set the ini file parameter startupDCscan toTRUE so SmilerShell rescans your DC information every time it starts.

Maybe you have some other program called DC that you'd like to run? Since SmilerShell's DC acts like an alias, you can bypass it by starting the command line with an equals sign.

Related Topics:
The Initialization File

Command Stack Files

If you have a set of commands you'd like to be able to load into SmilerShell, create a command stack file. This is simply an ASCII file with one command per line. By default, the command stack file name is "smishell.stk" and it is in the same directory as the ini file. However, you can use any name, location, or extension you like.

You can load a command stack file automatically when you start SmilerShell by putting the file's name in the cmdfile= line of your ini file. Command stack files can also be loaded or saved at any time from the File menu. Alternatively, if you want to pre-load just a few commands when SmilerShell starts, you can use the cmdstack= ini file parameter to list the commands right in the initialization file itself.

Related Topics:
The File Menu
The Initialization File

The Initialization File

You can initialize SmilerShell by setting parameters in an initialization file. This is a plain-text file, so use Notepad or another ASCII text editor to edit it. A convenient way of getting to your ini file is to use the Edit Ini File item on the File menu. This sets you up to edit your initialization file in Notepad. If you started SmilerShell without an initialization file, you'll be asked if you'd like to create one, filled with reasonable defaults, before proceeding. Then it'll set you up to edit it in Notepad.

If you want SmilerShell's initialization file to be named something other than "smishell.ini" or be somewhere other than the same directory as the SmilerShell program, give that information on the command line. To do this, edit the SmilerShell icon's Properties using Program Manager's Properties dialog (it's under the File menu). Add a space after "smishell.exe", then the flag "/ini=" and the drive and directory in which to find the ini file, with no embedded spaces, as in the following example:

Command Line: c:\smishell\smishell.exe /ini=c:\dir1\subdir2\myfile.ext

If you don't specify otherwise (in the initialization file), SmilerShell looks for its DC information file and any startup command stack file in the same directory as the initialization file.

SmilerShell's initialization file is set up just like every other ini file in Windows. For each section, there's a section header in brackets, under which are entries that are set to values:

```
[SmilerShell Params]
AliasesCaseSens=TRUE
cmdfile=smishell.stk
cmdstack=dir \dos;type \autoexec.bat;copy \config.sys junk.tmp;
confirmexit=TRUE
dirfile=c:\smishell\smishell.dir
DOSinWindow=FALSE
overtype=FALSE
prevdir=<set by SmilerShell when exiting>
prevposition=<set by SmilerShell when exiting>
restoretime=400
scandrives=bcde
showapps=TRUE
showclock=FALSE
showdir=FALSE
showinactives=TRUE
showmenu=TRUE
showresources=TRUE
startupDCscan=FALSE
timeformat=12
```

titlebar=TRUE
topmost=TRUE
UseComspec=TRUE
winwidth=500 or PREV

[SmilerShell Aliases]
TypedAlias=Replacement
(F3)=Replacement
word = c:\winword\winword
(f2) = type \autoexec.bat
(F5)= sol
dirprog=dir c:\mydir\mysubdir\programs*.exe
(F12)= notepad \config.sys
etc...

[SmilerShell Apps]
1=dir *.txt,DIR Text Files,TRUE
2=chkdsk,Check Disk,FALSE
3=notepad c:\autoexec.bat,Edit My Autoexec File,TRUE
4=sol,Solitaire,TRUE

You don't have to have all of the parameters. They don't have to be in the order shown above. If you don't specify a parameter, the SmilerShell default is used for it. If you specify a parameter, you can comment it out by putting a semicolon at the beginning of its line.

[SmilerShell Params]: Generally, parameters are read at startup and again whenever you Edit Ini File. However, some parameters are only read at startup (see list). The defaults, and the parameter meanings, are:

AliasesCaseSens=TRUE Each command you submit is tested to see if it matches a string on your alias list. By default, this alias testing is case-sensitive. To change this, set AliasesCaseSens=FALSE.

cmdfile= File from which to pre-load commands. By default, no filename specified, so no commands are pre-loaded. The cmdstack= setting takes precedence over the cmdfile= setting. That is, if commands are listed in the ini file using the cmdstack= parameter, they will replace any commands read in from a file named with the cmdfile= parameter. This parameter is only read at startup.

cmdstack= You can list the commands to be pre-loaded right in the ini file. By default, no commands are listed, so no commands are pre-loaded. Commands to be pre-loaded are all on one line, separated by a semicolon (see the example ini file above). The ini-file command list is a convenience feature. You can have up to 300 characters in this entry. If you need more commands loaded, use a cmdfile= file.

This parameter is only read at startup.

confirmexit=TRUE If this is TRUE (the default), you'll be asked to confirm that you really do want to exit from SmilerShell.

dirfile= File from which to load the DC information. SmilerShell creates this file the first time you use DC (with your permission), and whenever you Scan Directories. The default name is "smishell.dir" and the default location is the same directory as the SmilerShell ini file. Use dirfile= to give it a non-default name or location. A filename without a path is assumed to be in the same directory as the SmilerShell ini file. This parameter is only read at startup.

DOSinWindow=TRUE or FALSE; default is what the systemwide "_default.pif" flag is set to. Do you prefer to run DOS commands fullscreen or in a desktop window? If you don't specify, the current systemwide value, found in the file "_default.pif" in your Windows directory, is the initial setting. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the DOSinWindow parameter will be updated when SmilerShell exits. This parameter governs the initial setting of the DOS In Window item on the Options menu. It is temporarily toggled when you use the right-bracket command line flag.

overtype=FALSE By default, SmilerShell's command line is in insert mode, not overtype mode. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the overtype parameter will be updated when SmilerShell exits.

prevdir=<previous drive and directory> You never need to touch this parameter. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, SmilerShell creates the prevdir parameter and saves its current directory there at exit. Next time, SmilerShell starts up in this directory. If you've set winwidth to PREV but there's no prevdir parameter at startup, SmilerShell starts up in the program directory. This parameter is only read at startup.

prevposition=number, number, number, number You never need to touch this parameter either. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, SmilerShell creates the prevposition parameter and saves its current location there. The four numbers saved are the x, y, height, and width in screen coordinates. If you've set winwidth to PREV but there's no prevposition parameter at startup, SmilerShell chooses its location as if winwidth wasn't specified. This parameter is only read at startup.

restoretime=1000 How many milliseconds to pause before trying to regain the focus from an "(Inactive..." window. Windows needs a little pause here. How little can your system get away with? Default is one second (1000 milliseconds).

scandrives=c By default, when you Scan Directories only the C drive is scanned for directory names to be used with SmilerShell's DC command. DC can change to endpoint directories on other drives. Here is where you tell it what drives you want it to be aware of.

showapps=TRUE By default, the Apps top-level menu item is displayed. You can turn it off to save screen space. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showapps parameter will be updated when SmilerShell exits. This parameter governs the initial setting of the Options menu's Apps Menu item.

showclock=FALSE By default, the hour:minute clock on the title bar is not displayed. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showclock parameter will be updated when SmilerShell exits. By default, SmilerShell's clock uses a 12-hour (am/pm) format. To use a 24-hour format, set timeformat=24.

showdir=TRUE By default, the current directory is displayed as part of the SmilerShell window's title. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showdir parameter will be updated when SmilerShell exits.

showinactives=default is TRUE or FALSE, what the systemwide "_default.pif" flag is set to. After a DOS command ends, do you want to see the inactive results window or just have it vanish? If you don't specify a preference, the current systemwide value, found in the file "_default.pif" in your Windows directory, is the initial setting. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showinactives parameter will be updated when SmilerShell exits. This parameter governs the initial setting of the Inactives Stay Visible item on the Options menu. It is temporarily toggled when you use the asterisk command line flag.

showmenu=TRUE By default, the menu bar is displayed. If you've set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showmenu parameter will be updated when SmilerShell exits.

showresources=FALSE By default, resources are not displayed. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the showresources parameter will be updated when SmilerShell exits.

startupDCscan=FALSE By default, SmilerShell only scans when you tell it to, either by choosing the Scan Directories menu item or with DC's "/r" flag. If the directory layout of your system is constantly in flux, you may want to set the ini file parameter

startupDCscan toTRUE so SmilerShell rescans your DC information every time it starts. This parameter is only read at startup.

timeformat=12 By default, SmilerShell's clock uses a 12-hour (am/pm) format. To use a 24-hour format, set timeformat=24.

titlebar=TRUE By default, the title bar is displayed. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the titlebar parameter will be updated when SmilerShell exits. This parameter is only read at startup.

topmost=FALSE By default, SmilerShell is not always on top of other windows. If you have set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time, the topmost parameter will be updated when SmilerShell exits.

UseComspec=TRUE By default, SmilerShell uses the command processor listed in the COMSPEC environment variable. This is usually DOS's COMMAND.COM, but some people use alternate processors such as 4DOS or NDOS. To have SmilerShell explicitly use COMMAND.COM, set UseComspec to FALSE. This parameter is only read at startup.

winwidth=number or PREV The initial width of the SmilerShell window, in Windows device units. If you don't specify a winwidth, SmilerShell sets it wide enough to show the menu items. This works out to about 600 Windows device units if showapps and showresources are TRUE, 300 if they are both FALSE. You can also set winwidth to PREV, indicating that you want SmilerShell to come up next time with the same location and status as it ended this time. If you have, then at exit the system updates the parameters DOSinWindow, overtype, showapps, showclock, showdir, showinactives, showmenu, showresources, titlebar, and topmost, and saves its current location and directory as the prevposition and prevdir parameters. If you've set winwidth to PREV but there's no prevposition or prevdir parameter at startup, SmilerShell chooses its own location and startup directory as if winwidth wasn't specified.

[SmilerShell Aliases] are specified one per line, in the form:

alias=replacement

The alias-part is a single word, with no embedded spaces. Function-key aliases have the key name (F2 through F12) in parentheses.

You can define up to about 100 type-in aliases, depending on their length. You can define one function-key aliases for each of the keys F2 through F12.

For either kind of alias, the replacement-part can be any number of words, anything you can type on one line, up to the DOS limit of 128 characters per submitted command.

By default, aliases are case-sensitive. You can change this by setting AliasesCaseSens to FALSE.

Aliases can reference other aliases, but be careful to avoid self-referencing infinite loops (alias 1 defined in terms of alias 2 which is defined in terms of alias 1 which is ...).

[SmilerShell Apps]: This is the list of up to 75 "favorite applications" that is displayed when you select the Apps menu item. You should never need to edit this section by hand. It's much easier to use the Apps menu item Edit Apps List. The apps are in the form:

N=Command, MenuLabel, RunNow

N is a number; the commands are numbered 1, 2, 3, 4, etc. Command is the actual command to submit; it can use aliases, command-line flags (*>=), etc. MenuLabel is the text displayed on the menu. RunNow controls what to do when the item is selected from the menu. It is either TRUE to submit the Command immediately, or FALSE to select the Command into the commandline for editing.

Related Topics:
Aliases
Command Stack Files
The Apps Menu

Why Is This A Shell?

The word shell is sometimes used for a wrapper that surrounds another application and hides it. SmilerShell is the opposite of that. It makes all the power of the command line available from an environment in which that power is not otherwise accessible. But since it makes things more visible, rather than less visible, why is it called a shell?

It's a shell in another sense. Maybe you've seen programs that let you "shell out" to DOS, for example WordPerfect's Ctrl+F1 command, Shell. When you "shell out" it's like having a window into another environment, a pathway to a different level of functionality. That's what SmilerShell is, and that's why it's a shell.

Notices

VERSION: SmilerShell version 2.1

SYSTEM REQUIREMENTS: Requires Microsoft Windows 3.1

SOFTWARE LICENSE: Anyone is welcome to distribute unregistered copies of SmilerShell, in its entirety as distributed with this file, as long as none of the files in this package are modified or deleted.

Registered copies of SmilerShell may not be distributed. Only one user is authorized to use the program, on one computer. It may not be used in a multi-user setting without first obtaining a site license. It may be duplicated only for the purpose of making a reasonable number of backup copies.

DISCLAIMER: The author of this software package, Barry Smiler, has used his best efforts in producing this software and documentation. These efforts include the research, development, and testing of the software, and production of the documentation.

WARRANTY: The author makes no warranty of any kind, expressed or implied, with regards to the software or the documentation. The author shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of this software package.

PRICING: All prices subject to change without notice.

ASP STATEMENT: This program is produced by a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please write to the ASP Ombudsman at 545 Grover Road, Muskegon, MI 49442-9427 USA, FAX 616-788-2765 or send a Compuserve message via Compuserve mail to ASP Ombudsman 70007,3536.

CONTACTING THE AUTHOR: Barry Smiler, the author of SmilerShell, can be contacted through Compuserve email (72340,375), Internet email (72340.375@compuserve.com), U.S. mail (Bardon Data Systems, 1023 Key Route Blvd., Albany CA 94706), or phone (510-526-8470). In addition, the ASP maintains current contact information for all its members.

Uninstalling SmilerShell

SmilerShell tries to be considerate of the rest of your Windows system. It does not put any file in any directory other than the directory you've installed it into. To purge it from your system, simply delete the SmilerShell icons from your Windows desktop and delete the SmilerShell files from the directory you put them in. If you have put the SmilerShell ini file in another directory, delete it. If you have used SmilerShell's DC command, delete the DC info file. By default it is named "smishell.dir", and is in the same directory as SmilerShell's ini file.

Related Topics:
Installing SmilerShell

Installing SmilerShell

SmilerShell includes the following files:

smishell.exe	the program
smishell.hlp	the documentation, in Windows Help format
readme.txt	overview and installation instructions
install.exe	automated SmilerShell installer
sample.ini	sample initialization file
sample.stk	sample command stack file
file_id.diz	45 column x 10 line text description, for BBS uploads
vendor.doc	gives distribution permission
invoice.wri	registration invoice (direct to Bardon Data Systems)
register.txt	order toll-free and through distributors worldwide
whatsnew.txt	summary of new features
demopro.exe	demo of SmilerShell Pro

You can install SmilerShell automatically, using the enclosed auto-installer. To do this, simply run install.exe from within Windows. You can run it using File Manager, or the Run item on Program Manager's File menu, or in whatever other convenient way you choose. Give it the directory to put SmilerShell's files into, and the Program Manager group name for the SmilerShell icons (appropriate defaults are suggested). It'll do the rest.

If you prefer, you can install SmilerShell manually:

1) Copy the files to a convenient directory on your hard disk. (Actually, only smishell.exe and smishell.hlp need to be in this directory. The others are not required to run SmilerShell.)

2) Put the SmilerShell icon into a program group. To do this, bring up File Manager and set it to the convenient directory you chose in the previous step. Then drag-and-drop smishell.exe into your favorite Program Manager group. The SmilerShell icon should appear there.

(How to drag-and-drop: While the mouse is pointing at the word "smishell.exe" in File Manager, press and hold the left mouse button. While continuing to hold the button down, move the mouse to point into your favorite Program Manager group. Let go of the button. The SmilerShell icon should appear in the group.)

Optionally, you can set up an initialization file. See The Initialization File. But if you start without an initialization file, SmilerShell will offer to create one and fill it with reasonable values, then let you edit it in Notepad before proceeding.

Optionally, give a non-default location for the initialization file. See The Initialization

File.

Optionally, you can create a command stack file, a list of commands you want pre-loaded into SmilerShell when you start. See Command Stack Files.

Related Topics:
Uninstalling SmilerShell

Shareware is a way of distributing software. You get to try the software in your own environment for a limited period of time and decide if it meets your needs. If you like it and continue to use it, you are obligated to pay for it by sending in the registration payment.

Often, shareware is distributed through a vendor who may charge a small fee. The vendor's fee pays for copying and distributing the disk but does not pay for the software itself. After trying the software, you have the opportunity to decide if you want to pay for it. Your payment gives you the right to continue using the software.

The Register Menu Item

Until you register, SmilerShell 's menu bar includes a Register menu item, which provides information on how to register. Of course, after you register, this menu item is superfluous, and goes away.

Related Topics:
Shareware: Try Before You Buy

Quick Start / Hints And Tricks

SmilerShell is the ultimate Windows command line. It's just like shelling out to DOS. You type a command and press ENTER. It supports pipes, redirection, and internal DOS commands (and runs Windows programs too). It works just like the DOS command line. But SmilerShell is the best command line you've ever seen, as if the plain-vanilla DOS prompt was enhanced by lots of handy utilities. Here's how to get the most out of it.

Instant Install: You can install SmilerShell automatically, using the enclosed installer. To do this, simply run install.exe. You can run it using File Manager, or the Run item on Program Manager's File menu, or in whatever other convenient way you choose. Give it the directory to put SmilerShell's files into, and the Program Manager group name for the SmilerShell icons (appropriate defaults are suggested). It'll do the rest. The installer will make no changes to your system setup. It just copies files to the directory you specify, and adds icons to the Program Manager group you specify. If you choose to uninstall, just delete these files and icons.

Favorite Applications: List your favorite applications under the Apps menu item. Then just click on one to either select it into the command line, or run it straight off (you can set it up either way).

Command History and Search: Every time you run a command, it is saved on the command stack. To find a previous command of interest, type the first letter or two of that command, then press the up or down arrow key. Up-arrow searches back, down-arrow searches forward. It's a circular buffer, the last command is connected to the first, so you can search in either direction. You can save the current command stack to a file and reload it automatically at startup, or at any other time. This gives you a preloaded batch of commands you can search on. The startup loading is set up in the ini file. A full command history list is available from the File menu.

Command Line Editor: A retrieved previous command, or anything else you type, can be edited to suit. Think of SmilerShell as a one-line word processor. It supports insert mode, overtype mode, and clipboard cut/paste.

Aliases: When you press Enter, the first word of the command is compared to the alias list. If it matches, the alias is substituted for that first word. You can skip the alias testing by starting the command with an equals sign (for example =dir). You can also hang aliases off function keys F2 through F12; hit the key and the command runs. Both kinds of aliases are set up in the ini file.

Quick Directory Change: Type DC and the first few letters of the directory you want to be in. If it's unambiguous, boom, you're there, otherwise a list box pops up with the first possible match highlighted. If you haven't used DC yet, you'll be asked for permission to scan the drives listed in the DC ini file parameter. If you've scanned more than one drive, DC changes drive as well as directory if necessary to get you

where you want to go.

Get Small: SmilerShell has a very small window, but you can make it even smaller. Use the Options menu to get rid of the menu and title bar. Or type Alt-M to toggle the menu, Alt-L to toggle the title bar. Then mouse SmilerShell as small as you like.

SmilerShell Never Forgets: In the ini file, set winwidth=PREV and SmilerShell will start up next time in the same directory, in the same screen position, and with the same settings, as when you shut it down this time.

Change Ini File Settings On The Fly: Hit the Edit Ini File item on SmilerShell's File menu. It'll fire up Notepad with your ini file (and create one first if needed). When you're done editing and you close Notepad, SmilerShell will know. It'll read in the ini file and reset itself as indicated there.

DOS In A Window: Do you prefer to have DOS commands run fullscreen or in a window? Toggle this on the fly with the Options menu's DOS In Window item. Or to run one command as if DOS In Window is set to the opposite of its current value, start that command with a right-bracket (for example >dir \dos).

Inactives Stay Visible: After you run a DOS command, do you want the command's inactive window to stick around, or immediately vanish? Toggle this from the Options menu. Or to run one command as if Inactives Stay Visible is set to the opposite of its current value, start that command with an asterisk (for example *copy *.* b:).

Remove Inactives: Too many inactive windows cluttering your screen? Get rid of 'em with this Edit menu item, or just type Alt-R from the keyboard.

Clock: Toggle the titlebar clock from the Options menu, or just type Alt-C. Prefer 12-hour or 24-hour time? Use the timeformat ini file parameter.

Work With Files: The built-in command SHOW is often a useful alternative to DIR, since SHOW's file list lets you click on a filename to select it into the command line.

File Associations: With File Associations simply type in the filename without the program name, and quite often SmilerShell Pro will know which program to run. For example give "FILENAME.WRI" and SmilerShell knows to run Windows Write on this file. This works very nicely with SHOW: use SHOW to pick a file, then press Enter to run the proper program with that file.

Current Drive/Directory: Toggle this onto the titlebar from the Options menu, or just type Alt-D.

System Resources: Toggle the System Resources display onto the menu bar from the Options menu, or just type Alt-S, to see a real-time running report of your available Windows memory and resources.

Insert Or Overtype Mode: Toggle this from the Options menu, or just type Alt-O. In overtype mode a flag appears in the title bar.

Topmost Window: Make SmilerShell a "topmost" window from the Options menu, or just type Alt-T. That way, it's always visible and ready for use, even when you're working in another window.

Handy Configurations: Make SmilerShell topmost, turn on the clock, turn off the menu, mouse it as small as it goes (about as big as two icons) and stick it in the corner. The clock shows, and it's always ready for action. Or turn off the title bar, and mouse it even smaller. To move it on the screen without the titlebar, click the right mouse button in the edit area and hold it down while you move SmilerShell. Or make SmilerShell topmost and maximize it. When maximized, it only takes up the top line of your screen, not the whole display.

DOS Commands: Fullscreen Or Windowed

The Options menu item DOS In Window (Alt+W) controls whether active DOS commands called from SmilerShell run fullscreen or in a window. Use the initialization file parameter DOSinWindow= to set this to your preference at startup. If you don't specify a preference, then at startup SmilerShell sets the menu toggle to match the current value of the "_default.pif" flag.

To toggle DOS In Window= for just one command, start the command line with a right-bracket (example: >dir \dos). You can use this with the equals or asterisk flags if you want to (for example =*>dir).

Related Topics:
Submitting Commands
The Options Menu
The Initialization File

SmilerShell Pro

SmilerShell Pro makes the ultimate command line for Windows even more useful. It does everything that regular SmilerShell does, and in addition adds these features:

Press The Button: SmilerShell Pro takes up no space on your desktop. None. That's because it's usually a tiny button that hops unobtrusively into the titlebar of whichever application is currently active. Press the button to bring up the command line window. Click the SmilerShell menubar's Hide item (or type Alt+H) and the command line window vanishes again. And try this to save time: right-click the titlebar button, and the commandline appears with the Apps menu already displayed, ready for you to click on one of your listed apps and run it. It's the fastest way to start programs!

Multiple Commands On One Line: You can type multiple commands on one command line. Hit Enter and they are submitted in order. And unlike everything else in Windows, SmilerShell makes sure the previous command ends before it starts the next one.

Runtime Parameters In Aliases: Runtime parameters (%1, %2, etc.) make it easier to tell aliases what to do when you run them. And if you alias multiple commands onto one line, you can make it act almost like a batch file, all within SmilerShell!

Change Your PATH: By default SmilerShell Pro uses the DOS search path. You can change this by typing a PATH command, just like in DOS. Later, go back to the old DOS path by typing PATH= (an equals sign but no parameters). PATH by itself shows the current path.

A Tiny Window: Want the smallest possible SmilerShell? With SmilerShell Pro, if System Resources is enabled and you toggle off the menu, the resources report will appear in the command line. Press any key and your command line returns. The cursor is where you left it, selections are still selected, and the key you pressed is typed into place in the command text. Turn off the titlebar as well as the menu for an incredibly small SmilerShell, with no loss of information!

It's Fast: Because of special software technology, SmilerShell Pro runs commands faster than SmilerShell Standard Edition.

Like SmilerShell Standard Edition, SmilerShell Pro come with a printed manual, support by phone, mail, or Compuserve, and these Extra Free Bonuses: two more handy Bardon utilities (WHATSNEW lists files not yet backed up, or directories containing such files; PR/PRFILTER format output for printing, and add a header with filename, size, create date, and print date), discount certificate for JCSM shareware CD-ROMs at half price, discounts on PsL shareware-by-mail (up to 2/3 off!), free Compuserve startup kit, and a disk-full of top Windows shareware I think you'll like.

Curious about everything, for no good reason, eh?

I like that!

Okay, how about this -- you are now eligible for Special Offer XYZZY, which entitles you to a !0% discount on SmilerShell ($17.95 plus shipping) or SmilerShell Pro ($26.95 plus shipping). Do this direct through me, at the Bardon address, okay? The distributors can't handle this sort of thing.

Be sure to mention Special Offer XYZZY when you order.

-Barry

SHOW: An Alternative To DIR

The built-in alias SHOW is sometimes more useful than DIR. Like DIR, you could issue, say, the command SHOW *.COM to display all COM files in a list. But unlike DIR, SHOW lets you choose one of the files from its list. The filename you choose is placed on SmilerShell's command line, where it becomes the current in-process command. You can add any needed parameters before submitting it.

SHOW works nicely with File Associations. Because SmilerShell supports File Associations, quite often you only have to give the filename, without naming the program to run it. SmilerShell can often tell what program to run, just by looking at the filename. So, for example, you could use SHOW to select "FILENAME.WRI" into the command line, then just press Enter. SmilerShell knows to run Windows Write on this file.

Maybe you have some other program called SHOW that you'd like to run? Since SmilerShell's SHOW acts like an alias, you can bypass it by starting the command line with an equals sign.

Keyboard Shortcuts

These SmilerShell commands work immediately, without going through a menu.

Alt+C	Clock in titlebar
Alt+S	System Resources report in menubar
Alt+D	Current directory in titlebar
Alt+O	Overtype/insert mode
Alt+T	Topmost window
Alt+L	Show/hide titlebar
Alt+M	Show/hide menubar
Alt+I	Inactive windows stay visible
Alt+W	DOS commands windowed/fullscreen

You can display any menu from the keyboard. When in a menu, press the underlined key in a menu item to run that item.

Alt+F	File menu
Alt+E	Edit menu
Alt+N	Options menu
Alt+A	Apps menu
Alt+P	Help menu

These standard Windows commands will work in SmilerShell.

F1	Help
ALT+F4	Exit
Alt+Bksp	Undo last action
Ctrl+Z	Undo last action
Shift+Del	Cut selected text, send to the Clipboard
Ctrl+X	Cut selected text, send to the Clipboard
Ctrl+Ins	Copy selected text, send to the Clipboard
Ctrl+C	Copy selected text, send to the Clipboard
Shift+Ins	Paste contents of Clipboard into commandline
Ctrl+V	Paste contents of Clipboard into commandline
Del	Clear selected text
Esc	Clear entire commandline
Ctrl+Esc	Show task list of all running programs

The Apps Menu

You can list up to 75 of your favorite applications under the Apps menubar item, so you can run one by clicking on it. To add, change, or delete a list entry, open the Apps menu and choose Edit Apps List. To edit an existing item, select it from the list. The command's information appears in the editing area. Click the Delete App button to delete it, or edit the information and press Change App to change it. To add a new command, give information for it and click on Add App. Give the menu text and command name. Choose whether, when its menu item is clicked, this command should be copied into the SmilerShell window for editing, or run immediately. It will be added after the currently-selected command, or at the end of the list if no command is currently selected. Commands listed here can use aliases, commandline flags (*>=), or any other valid options.

There is a Browse button available. Click it to display a dialog box with which you can navigate through your system and find the program name to be used on the command line. In that box, click on a filename to select it into the Command to run or fetch line.

The Options menu item Apps Menu controls whether the Apps top-level menu item is shown on the menubar. If necessary, you can remove the Apps item to save space. The ini file parameter showapps= controls the initial setting of the Options item Apps Menu.

Related Topics:
The Options Menu