IMPOSTER

Version 1.01
Copyright (c) 1991-1992
Darrell W. Burgan
All Rights Reserved

IMPOSTER

Chapter 1 - Introduction

Thank you for your interest in IMPOSTER. This is my first publicly available program for Microsoft Windows; I hope you find it a useful addition to your library of Windows applications.

Please read the following sections carefully, as they provide important information about using and distributing IMPOSTER without violating my copyrights.

Freeware

If you are unfamiliar with freeware, it is simply an alternative distribution channel for quality software. You are allowed to use and distribute freeware products without restriction.

Freeware is not public domain, however. Public domain software generally places the source code as well as the executable product into the hands of the public. Freeware, however, places only the executable. The author still retains copyrights on the source code. This means that though you are free to use and distribute a freeware product, you cannot modify it, sell it or use it in any other way except how the author intended it.

Terms of Use and Distribution

IMPOSTER is offered as freeware. This means you are free to pass it on to your friends, to post it on bulletin boards and other online services or otherwise distribute it as long as you abide by the following terms:

- You distribute all files associated with IMPOSTER *exactly* as you initially received them. The only way to ensure this is to distribute only the archive it initially came in.

- You do not modify, reverse engineer, add to, or otherwise change the product. If you want to develop enhancements or discover problems with the product, please contact me as described in the technical support section.

- You do not distribute IMPOSTER in conjunction with any product for which anyone will derive profit without my prior written consent.

- You do not charge for the distribution of IMPOSTER, except for a nominal disk-handling fee not to exceed five U.S. dollars per disk.

Also, please read the following notice carefully:

> *THIS PRODUCT, CALLED IMPOSTER, IS OFFERRED ON AN AS-IS BASIS. NO WARRANTIES OF FITNESS OR MERCHANTIBILITY, EXPRESS OR IMPLIED, ARE GIVEN. NO OTHER WARRANTIES OF ANY KIND ARE GIVEN.*
>
> *UNDER NO CIRCUMSTANCES WILL THE DEVELOPER, DARRELL BURGAN, BE HELD LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL OR ANY OTHER DAMAGES CAUSED BY ANYTHING THIS PRODUCT DOES OR FAILS TO DO.*
>
> *THE USER BEARS THE FULL RESPONSIBILITY FOR VERIFYING THIS PRODUCT IS RELIABLE AND WORKS TO THEIR SATISFACTION.*
>
> *IF YOU CANNOT ACCEPT THESE TERMS OR ANY OTHER TERMS SPECIFIED IN THIS DOCUMENT, PLEASE DO NOT USE THIS SOFTWARE. BY USING THIS SOFTWARE, YOU INDICATE YOU AGREE WITH THESE TERMS AND WILL ABIDE BY THEIR RESTRICTIONS.*

Sorry to offer the product under such terms, but I am a one-man development team and simply do not have the resources to adequately warrant this product. Since I am not charging you for the use of this product, I don't think this is unfair.

Still, I have to the best of my abilities verified that this product is substantially bug-free and will work on standard Windows platforms. I just cannot guarantee it will work properly under all circumstances or that any bugs haven't escaped my attention.

Registration

In return for allowing you to use IMPOSTER free, I simply ask that you "register" your copy. Unlike with shareware, registering IMPOSTER simply means you are letting me know you are using the product, are interested in other products that might be released, and agree to the terms of use and distribution. There is no cost whatsoever associated with registering IMPOSTER. When you register:

- You will be placed on my electronic mailing list and informed of any new products/versions prior to public release. I may also attempt to keep anyone notified via U.S. Post that has no access to an electronic mail service, but I cannot guarantee this due to the potential cost.

- And you will encourage me to continue to develop useful tools for Windows and make them publicly available via freeware.

I strongly encourage everyone who uses IMPOSTER and finds it useful to register, just so I can keep track of the success of this product.

To register, simply photocopy the following page, fill out the form, sign the form (indicating you agree to abide by the terms of use and distribution) and send it to the specified address.

If you like, you can also register electronically by sending me email via any of the three electronic technical support mechanisms. Just be sure to include the same information as on the form. REMEMBER: If you register electronically, you are implicitly agreeing to abide by the terms of use and distribution and accept their legal restrictions.

Thanks in advance, if you decide to register.

Imposter Registration

Send to: Darrell Burgan
6400 Independence Parkway Suite 4103
Plano, TX 75023

Name: _____
Company: _____
Address: _____
City: _____
State: _____
Zip: _____

Product Registered: **IMPOSTER**
Version Registered: **1.01**

IMPORTANT: I have read the terms of use and distribution and agree to abide by their terms and restrictions.

Signature: _____ Date: _____

Comments:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Suggestions for Enhancements:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Technical Support

If you should encounter problems, if you have suggestions for improvements, or if you just want to talk Windows, you can contact me any of the following ways:

- The Digital X-Connect BBS - 214-517-8443. This is the official distribution site for IMPOSTER and the preferred mechanism for contacting me. Just send mail to the SYSOP, to ALL, or to Darrell Burgan in the IMPOSTER Conference and I'll be there to help you!

- CompuServe - Send mail to 73020,1023.

- America Online - Send mail to DarrellB.

- US Post. Send it to the following address:

Darrell Burgan
6400 Independence Suite 4103
Plano, TX 75023

Since electronic correspondence is much quicker and less expensive, I strongly discourage regular mail unless you simply don't have access to a modem. I also cannot guarantee responses to messages sent this way.

I will provide technical support and answer questions to anyone using any of my products. One of the things I have learned in my years in the computer business is that good support is often worth as much as the benefit of using a good product. It is my goal to provide the best possible technical support to all users of my products. If your question may not be directly related to one of my products, you should still feel free to contact me and I will help you to the best of my abilities.

I will attempt to respond to all queries within 24 hours of receipt, but cannot guarantee that this will always be possible (I go on vacation every now and then, too). If you have an urgent question, your best bet is to send the question via all three electronic mail services listed above.

I am not currently offering voice technical support. If, however, you need a voice phone number to be able to register the product, contact me via email and we can make appropriate arrangements.

Credits

Someone once said we stand on the shoulders of those who came before us, and I am certainly no different.

Many, many thanks are due to:

- Andy Walding of the Digital X-Connect BBS for providing a support mechanism for IMPOSTER and outstanding advice on advertising and distributing this product. If you're looking for a good technical BBS with active file and message bases, look no farther--the X-Connect is for you.
- Neil Rubenking and the members of Team Borland on CompuServe for their outstanding TPW advice and assistance.
- Borland International for making Turbo Pascal a reality under Windows. If you've been writing Windows apps using the usual Microsoft tools, take a look at TPW--once you've tried it, you'll never go back.

Dedication

Last I want to dedicate this product to my son Reese Gilbert, who is the source of much of my inspiration, joy, and heartburn (after all, he's in his terrible two's). Children are truly a gift from above.

Chapter 2 - IMPOSTER

What is IMPOSTER and why did I create it? Put simply, IMPOSTER is a DOS command interpreter that attempts to closely mimic COMMAND.COM. Using IMPOSTER allows you to use Windows as a superior operating system but retain all the familiarity and utility of the DOS command line.

IMPOSTER supports to some degree the entire internal DOS command set, as well as most of the external commands defined in DOS 5.0. When you're using IMPOSTER, it's easy to forget that you're NOT running a DOS program at all--just a carefully designed Windows application that emulates DOS as closely as possible.

Why Use IMPOSTER?

Why not just run COMMAND.COM in a window? Well, that works okay if you can run Windows in 386 Enhanced Mode, but some of us poor grunts are still trying to use our 286-class machines with Windows. IMPOSTER works just as well in Standard Mode as in Enhanced Mode, and IMPOSTER can even run in Real Mode for 8086-class diehards.

IMPOSTER also offers a lot of features that just can't easily be duplicated from running COMMAND.COM in a window:

- Is optionally able to run DOS *or Windows* applications from the command line, just like in DOS. Can execute .PIF files in exchange for .EXEs or .COMs if configured properly. COMMAND.COM simply can't do this at all.

- Fully supports the WIN.INI file extension associations to allow you to bring up an application just by typing the name of one of its documents. Again, COMMAND.COM just cannot accomplish this.

- Provides a much-needed mechanism for running batch files under Windows. When IMPOSTER runs a batch file, it can run Windows apps along with DOS apps, can execute any of the IMPOSTER internal commands, can utilize IMPOSTER's extended batch commands, etc. COMMAND.COM runs DOS batch files, accesses DOS apps and that's about it.

- Greatly reduced memory consumption. Now you can afford to have a DOS command-line available at all times, even while running other memory-hungry applications. You can even run several instances of

the command line simultaneously without eating megabytes of RAM. And Windows can actually "borrow" memory from IMPOSTER if space gets tight. COMMAND.COM eats a minimum of 256K of RAM, for every single instance, and Windows can't get any of it back until you exit COMMAND.COM!

- Superior multitasking. Since IMPOSTER is a true Windows application, it doesn't depend on the somewhat risky time-slice multitasking of DOS windows in 386 enhanced mode. Ever try to run two or three DOS windows simultaneously? Not only does your whole system slow down, but strange things begin to happen if the system is really pushed hard. IMPOSTER is able to detect when things are going on in other processes and wait until the system is idle to continue processing. The result? Much smoother and safer multitasking.

- Program Manager replacement. IMPOSTER can serve as a fully functional Windows shell, replacing the Program Manager at startup. For DOS fanatics like me, this makes Windows' pretty-but-not-very-efficient interface more than just merely tolerable. This of course is impossible with COMMAND.COM.

- Supports nearly every internal DOS commands, several external DOS commands, several of the DOS Burgan Utilities commands and several Norton Utilities commands. You need the Norton Utilities to do this with COMMAND.COM.

Being a command-line fanatic myself, I feel IMPOSTER offers the best of both worlds--a DOS command line for speed and Windows for access to graphics and superior applications. All those icons just clutter up the screen, in my view. Of course, you can use the Program Manager or any other shell, if that is your preference, and use IMPOSTER simply as a regular Windows application.

Installation

Installing IMPOSTER is a fairly simple task:

- First make sure you have the IMPOSTER.EXE, IMPOSTER.ALS, IMPOSTER.WRI and IMPOSTER.HLP files available. It is also generally a good idea to scan any files for viruses before you install them on your PC, if you have not done so already.

- Copy IMPOSTER.EXE, IMPOSTER.ALS, IMPOSTER.WRI and IMPOSTER.HLP

to any directory on your path. The Windows directory will work fine. NOTE: *IMPOSTER must be on your path to work properly*, especially if you intend to use IMPOSTER as the Windows shell!

- Last, run IMPOSTER.EXE and bring up the Preferences dialog box. Make any changes you like, then press the "OK" button. This will make the following changes for you automatically:

    - The default order of the PROGRAMS= line in your WIN.INI is not very convenient for IMPOSTER use. IMPOSTER changes them from:

      Programs=com exe bat pif

      To:

      Programs=pif exe com bat

    - If you wish to use IMPOSTER as your Windows shell, IMPOSTER will edit the SYSTEM.INI file and change the SHELL= line from:

      Shell=PROGMAN.EXE

      To:

      Shell=IMPOSTER.EXE

      It will also add a new line to the [IMPOSTER] section of WIN.INI:

      OldShell=[shellname]

      IMPOSTER uses this to put your old Windows shell back if you choose for IMPOSTER to no longer be the Windows shell.

    - IMPOSTER will add the following line to the [Extensions] section of your WIN.INI:

      IPS=IMPOSTER.EXE ^.IPS

      Adding this line is important because it allows you to run IMPOSTER batch files from the LOAD= or RUN= lines of your WIN.INI, even when IMPOSTER is not the shell. If

IMPOSTER is the shell, then you are free to run .BAT or .IPS scripts from WIN.INI.

- IMPOSTER will finally add the following lines to your WIN.INI at the bottom:

```
[IMPOSTER]
    ForegroundColor=GRAY
    BackgroundColor=BLACK
    Font=OEM
    StartupCommand=
    HorizontalSize=80
    VerticalSize=25
    InitialFocus=1
    ConfirmWindowsExit=1
    LastShell=PROGMAN.EXE
    MinimizeOnUse=0
    IgnoreHotkey=0
    HotkeyIndex=20
    EnableHotkey=1
    ControlKey=1
    ShiftKey=0
    AltKey=0
    CommandDelimiter=^
    DefaultCmdShow=3
    UpperLeftX=0
    UpperLeftY=0
    WindowWidth=644
    WindowHeight=256
```

Depending on what preferences you entered, the values for these WIN.INI keywords may vary.

And that's all there is to installing IMPOSTER!

WIN.INI Settings

Just so you know what's going on, the following information documents the WIN.INI settings IMPOSTER uses.

HorizontalSize=xx

Specifies the width of the virtual screen in characters. The default is 80 if this setting is not supplied. The maximum allowed is 132. This does not affect the initial size of the IMPOSTER window, only how wide the screen is logically before line wrap occurs.

VerticalSize=xx

Specifies the height of the virtual screen in lines. The default is 25 if this setting is not supplied. Also, the maximum here is 256 as well. Again, this only affects the logical screen size (how many lines before scrolling occurs),

not the initial height of the IMPOSTER window. This can be a useful parameter to set fairly high (e.g. 99 lines), to allow you to see more of what you have recently done.

Also note that the horizontal size x vertical size cannot exceed 32767; if you attempt settings that exceed this limit, IMPOSTER will assume 80x25.

ForegroundColor=xxxxxxx

Specifies the foreground color to use in IMPOSTER. Possible choices are BLUE, RED, GREEN, PURPLE, CYAN, WHITE, YELLOW, GRAY, BROWN, BLACK or MAGENTA. The default color is GRAY. If you specify anything else, IMPOSTER will assume GRAY. Note that due to the way Windows maps colors to the output device, some colors may only be close to what you specify. For example, on some EGA monitors PURPLE is actually displayed as a shade of blue.

BackgroundColor=xxxxxxx

Specifies the background color to use in IMPOSTER. Possible choices are also BLUE, RED, GREEN, PURPLE, CYAN, WHITE, YELLOW, GRAY, BROWN, BLACK or MAGENTA. The default color is BLACK. If you specify anything else, IMPOSTER will assume BLACK. Again, color mapping is approximate due to the color mapping approach Windows uses.

StartupCommand=xxxxxxxxxx

Specifies a command that is always to be executed upon startup of IMPOSTER. For example, if you always want a prompt that contains the current time only, you could say "StartupCommand=PROMPT tg" and receive a prompt that looks something like "11:33:25.03>". Or you could say "WINDOW MAXIMIZE" and IMPOSTER will always start in the maximized state. You can also run batch files or any other IMPOSTER command from this line, allowing you to implement something similar to an AUTOEXEC.BAT for your IMPOSTER instances. Bear in mind, however, that the startup command is executed every time you start IMPOSTER, even if you run multiple instances.

LastShell=xxxxxxxxxx

Specifies what the previous Windows shell was, in case you no longer want IMPOSTER to be the Windows shell. If you never selected IMPOSTER to be the Windows shell, this line may not appear. If you delete this line and then

no longer want IMPOSTER to be the shell, IMPOSTER will warn you and assume that Program Manager is the new shell.

Font=xxxxxxxxxx

Specifies what font you wish to use in the IMPOSTER window. It is a good idea to use one of the OEM fonts unless you just plain hate them, since extended ASCII characters are represented properly in this font. If your batch files use extended ASCII characters and you select a different font, you may see international language characters instead of what you expect.

MinimizeOnUse=x

Indicates if IMPOSTER should minimize itself whenever another application is launched. If this parameter is set to 1, IMPOSTER will minimize itself upon a program launch; if this parameter is set to 0, it will not. This WIN.INI parameter is actually controlled through the "Minimize On Use" menuitem, not the Preferences box.

InitialFocus=x

Indicates if IMPOSTER should always bring the focus to itself upon startup. If the value of the parameter is 1, it will autofocus; if it is 0, it will not.

ConfirmWindowsExit=x

Indicates if IMPOSTER should ask before exiting Windows. If the value of the parameter is 1, it will ask; if it is 0, it will not.

EnableHotkey=x

Indicates if IMPOSTER should use a certain key combination as a "hotkey" to bring the focus to the IMPOSTER window. If the value of the parameter is 1, IMPOSTER will watch for a hotkey event; if it is 0, it will not.

IgnoreHotKey=x

Indicates if IMPOSTER should ignore the hotkey whenever pressed. If the value of the parameter is 1, it will ignore the hotkey; if it is 0, it will not. This switch has no meaning if hotkey processing is not enabled.

HotkeyIndex=x

Indicates which keystroek IMPOSTER will use as the hotkey. Refers to an

internal key table.

ControlKey=x

Indicates if the hotkey keystroke only is used if the <Control> key is pressed simultaneously. If the value of the parameter is 1, IMPOSTER will require <Control> to be pressed; if it is 0, it will not.

ShiftKey=x

Indicates if the hotkey keystroke only is used if the <Shift> key is pressed simultaneously. If the value of the parameter is 1, IMPOSTER will require <Shift> to be pressed; if it is 0, it will not.

AltKey=x

Indicates if the hotkey keystroke only is used if the <Alt> key is pressed simultaneously. If the value of the parameter is 1, IMPOSTER will require <Alt> to be pressed; if it is 0, it will not.

CommandDelimiter=x

Indicates what character will serve as the delimiter between multiple commands on a command-line. The default is the caret character ('^').

DefaultCmdShow=x

Indicates the default initial state for applications launched by IMPOSTER. A '0' will launch applications hidden, '1' will launch applications minimized, a '2' will launch applications normally and a '3' will launch applications maximized.

UpperLeftX=x UpperLeftY=x WindowWidth=x WindowHeight=x

These flags indicate the saved dimensions of the IMPOSTER window, in screen coordinates (pixels).

Console Control Keys

IMPOSTER supports a large number of console control and editing keys. In general, the following keys are always supported:

<Ctrl-C>	Cancels the command you just executed. Has no effect when the command-line is idle.
<Ctrl-S>	Pauses screen output of any command. Also has no effect when the command-line is idle.

<Ctrl-Q> Resumes screen output following a <Ctrl-S>. Has no effect otherwise. Note that after pressing <Ctrl-S>, any key except <Ctrl-S> will resume screen output.

Many additional keys are also supported when the command-line is idle and accepting input:

<Esc> Erases anything input and returns the cursor to the start of the input line.
<Right> Moves the cursor one character to the right.
<Left> Moves the cursor one character to the left.
<Ctrl-Right> Moves the cursor one word to the right.
<Ctrl-Left> Moves the cursor one word to the left.
<Home> Returns the cursor to the start of the input line.
<End> Moves the cursor to the end of the last word in the input line.
<Ins> Toggles insert/overstrike mode.
<Del> Deletes the character pointed to by the cursor.
<Backspace> Deletes the character to the left of the cursor.
<Enter> Submits the command-line to IMPOSTER for processing.
<Ctrl-Enter> Submits the command-line to IMPOSTER for processing; if your command is to launch another program, that program will start maximized. If your command is an internal IMPOSTER command, it has no other effect.
<Shift-Enter> Submits the command-line to IMPOSTER for processing; if your command is to launch another program, that program will start minimized. If your command is an internal IMPOSTER command, it has no other effect.

Several keys are also supported to implement command history/recall:

<F3> Recalls the last command you executed. If you have not executed a command yet, <F3> has no effect.
<Up> Recalls the most recent/previous command in the command history ring buffer. Pressing <Up> several times will show you the last several commands you executed, in reverse order.
<Down> Recalls the least recent/next command still stored in the command history ring buffer. Pressing <Down> several times will show you the commands you executed, in order. Note that the command history ring buffer can store only the last 16 commands you've

entered.

Last, the mouse is used as a "short-cut" to typing in a command by pointing to a word visible on the screen:

<Click> Copies the word to which you are pointing to the command-line. This can be useful, for example, for the COPY command.

<Double-Click> Copies the word to which you are pointing to the command-line and executes the command-line. This can be used to point to a filename on the screen and execute it by double-clicking.

Hot Keys

IMPOSTER also supports a single "hot key", which can be used to bring IMPOSTER into the foreground, then put it back into its previous state.

For example, if you are currently in your favorite word processor but need to quickly copy a file to a floppy, you simply need to hit the IMPOSTER hot key. This will pop IMPOSTER up over the word processor and allow you to enter whatever IMPOSTER commands you like. Pressing the hot key again will bring your word processor back into the foreground and put IMPOSTER back where it was before.

You can select the hot key IMPOSTER will use from the Hot Key menuitem. Note that if hot key processing is not enabled, IMPOSTER will not notice when hot key events occur. Hot key processing imposes a slight overhead on all Windows keyboard activities, even if you are in other applications, so it should be disabled if you do not plan to use this feature.

IMPOSTER also allows you to instruct it to temporarily "ignore" hot keys, even if hot key processing is enabled. This might be useful if you need to run an application that also finds the same hot key significant. This is toggled by the Ignore Hot Keys menuitem. Note that even if IMPOSTER is temporarily ignoring hotkeys, some slight overhead will still occur on all Windows keyboard activities.

Aliases

IMPOSTER supports aliases for any internal or external command. An alias is a name for a command that is commonly executed. When you execute an

alias, IMPOSTER simply substitutes the definition for the alias for the actual alias name.

This simple structure allows you to provide short-cut names for other commands. You can access the alias list by selecting the Aliases menuitem from the Options pulldown menu.

From the alias list dialog box, you can add, change or delete aliases and close the dialog box with or without saving your changes. When you click on the Add button, IMPOSTER will ask you for the alias name and definition for this alias. If you click on Change, IMPOSTER will only allow you to modify the definition. Clicking on Delete will remove the selected alias from the list.

If you click on OK, all changes will be saved to IMPOSTER.ALS, otherwise none of your changes will be saved. Note that if IMPOSTER.ALS does not exist, IMPOSTER will create it for you automatically with no aliases defined.

IMPOSTER is distributed with a predefined set of aliases in IMPOSTER.ALS. Feel free to modify or delete these definitions as you see fit.

Redirection and Piping

IMPOSTER supports redirection of command input and output as well as standard DOS piping. For instance, you may want to send the output of the "DIR" command to a file, or you may wish to see a long directory listing in sorted order with pauses after every screenfull. You can accomplish this via redirection and/or piping.

Redirecting Output

To redirect the output of a command to a file, simply use the greater-than character ('>'). For example, to send the output of the DIR command to the DIR.TXT file:

```
DIR > DIR.TXT
```

This will write the same text that ordinarily would appear on the screen to DIR.TXT, overwriting any DIR.TXT that may already exist.

If you want to append to the output file rather than overwrite, simply use a double-greater-than:

```
DIR >> DIR.TXT
```

If DIR.TXT doesn't exist, IMPOSTER will create it, otherwise the output will be appended to the end.

Redirecting Input

Redirecting input means that a command will receive its input from a file instead of the keyboard. This is most useful with a special type of command called a "filter". Filters generally take the contents of an "input stream", perform some manipulation, and write the output to the output device. By redirecting a file as the input to a filter, you are telling that filter to use the contents of the file as the input stream.

For example, say you had an ASCII file called PEOPLE.TXT and you wanted to sort it, displaying the output on the screen. This is easily accomplished with input redirection:

```
SORT < PEOPLE.TXT
```

PEOPLE.TXT will be sorted and the output displayed on the screen. Now say you wanted to sort the file but write the output to a new file called PEOPLE.OUT:

```
SORT < PEOPLE.TXT > PEOPLE.OUT
```

That's all there is to input redirection.

Piping

Piping can be viewed as simply a combination of input and output redirection, where the output of one command becomes the input to the next command. For example, say you wanted to see a sorted directory listing. You could accomplish this using simple redirection:

```
DIR > DIR.TXT
SORT < DIR.TXT
DEL DIR.TXT
```

The problem with this approach is that it quickly becomes ungainly. For example, now say you also want the sorted listing to pause after every

screenfull:

```
DIR > DIR.TXT
SORT < DIR.TXT > DIR.SRT
MORE < DIR.SRT
DEL DIR.TXT
DEL DIR.SRT
```

Wouldn't it be better to be able to say all this in one command? Piping was created for this very purpose. For example, to see a sorted directory listing:

```
DIR | SORT
```

This command will take the output of the DIR command and automatically redirect it as the input to the SORT filter. The resulting sorted listing will be displayed on the screen.

Now what if you want the output to pause after every screen? Try this:

```
DIR | SORT | MORE
```

This approach is a lot cleaner that specifying all the redirection yourself.

Note that you can also combine redirection with piping. For example, if you wanted to sort a directory listing but save the output in a file:

```
DIR | SORT > DIR.TXT
```

The sorted directory listing will be saved in DIR.TXT.

Command Set

As I mentioned previously, IMPOSTER supports all internal DOS commands and many external DOS commands and third-party utilities. The following is a list of all commands IMPOSTER responds to:

ATTRIB/ATTR

Format: ATTRIB [+/-R] [+/-S] [+/-H] [+/-A] [<path>] [/S] [/Q]

Putting a '+' before an attribute turns the attribute on. A '-' will turn the attribute off. The following are attributes that are changeable:

R - Read-only attribute
S - System file attribute
H - Hidden attribute
A - Archive attribute

If no attribute changes are specified (i.e. only a path is specified, like "ATTRIB *.*"), the current attributes of all files meeting the path are displayed. If no path is provided, ATTR assumes the current directory.

If /S is specified, ATTR will also search all subdirectories and apply any attributes changes to those as well.

If /Q is specified, ATTR will not echo any output to the screen.

BREAK

Format: BREAK

Displays the current status of the DOS BREAK flag. Does not allow you to change the status.

CDD

Format: CDD [[drive:] path]

Changes the current drive and directory to the specified drive and directory.

If the path is not specified, CDD displays the current drive/directory. If the path specified is only a drive letter ("C:"), CDD will switch to the current directory on that drive. If no drive letter is specified, CDD assumes the current drive.

CHDIR/CD

Format: CD [path]

Changes the current directory to the specified directory.

If the path is not specified, CD displays the current directory. If the path specified is only a drive letter ("C:"), CD will display the current directory on that drive.

CLS

Format: CLS

Clears the screen.

COPY

Format: COPY [/V] <path1> | CON <path2>

Copies file(s) from path1 to path2. The /V parameter indicates the source and destination files are compared byte for byte to ensure it was successful.

If you specify CON as the second parameter, IMPOSTER will read text from the console device until you hit <Ctrl-Z> and save what you typed to the file. If you hit <Ctrl-C> instead of <Ctrl-Z>, IMPOSTER will not save your changes.

Note that this implementation of COPY copies ALL files in binary mode, regardless of whether they are ASCII or not, unless you are executing COPY CON. This ensures <Ctrl-Z> characters don't cause premature EOFs.

COPY does not currently support concatenation of files.

DATE

Format: DATE [mm/dd/yy]

Updates the current DOS date. If no mm/dd/yy is specified, IMPOSTER will respond with:

```
Current date is Mon 03/12/1991 Enter new date (mm/dd/yy):
```

If the user hits enter at this prompt, the date is not changed.

DATECOPY

Format: DATECOPY <path> <directory>

DATECOPY simply copies all files matching the path to the directory, based on the file date/times.

DATECOPY accomplishes this by analyzing all DOS time stamps on the files and copying all later than the earliest to the destination directory. This is particularly useful with a "touch" utility; for example, simply "touch" the file(s) you wish to track. From that point forward, any file that is modified will automatically be copied by DATECOPY.

Why not use XCOPY and copy only things that have the archive bit set? Unfortunately, just reading a file sets the archive bit, whereas the date/time is only reset if the file is changed in some way. DATECOPY ensures that only files you have truly changed are copied.

DEJUNK

Format: DEJUNK

DEJUNK is an IMPOSTER filter, which will search the input stream for all non-standard ASCII characters and control characters, removing them from the input stream.

This could be used, for example, on an ASCII file captured from a communications session that had a large number of control characters in it.

Like all IMPOSTER filters, DEJUNK will accept input only from a redirected file or a pipe.

DETAB

Format: DETAB [<tab size>]

DETAB is an IMPOSTER filter, which will search the input stream for tab characters and replace them with sufficient spaces to emulate the tab stop.

The tab size tells DETAB how many characters are between tab stops. The default tab size is 8 characters if the tab size is not specified.

Like all IMPOSTER filters, DETAB will accept input only from a redirected file or a pipe.

DIR

Format: DIR [<path>] [/W] [/P] [/2] [/A] [/S] [/D] [/N] [/L] [/B]

Displays the directory specified or the current directory.

If /W is specified, only filenames will be displayed in 5 columns. If /2 is specified, full directories will be displayed in 2 columns. The /2 and /W switches cannot both be specified.

If /P is specified, the screen will be paused after the screen fills.

If /A is specified, all files will be shown regardless of attributes. If /A is not specified, only non-hidden and non-system files will be shown (as in DOS).

If /S is specified, DIR will show the contents of all subdirectories as well.

If /D is specified, only subdirectories will be shown.

If /B is specified, a 'bare' format will be used, with no header or summary information.

If /L is specified, all output will be shown in lowercase.

If /N is specified, a "compressed" form of the DIR command will be displayed. This compressed form is more suitable for use with the click and double-click features of the mouse in the IMPOSTER window. By default, DIR will take the uncompressed format that properly emulates DOS but is less friendly to the mouse.

If there is a DIRCMD environment variable defined and you specify DIR with no switches, IMPOSTER will append the contents of DIRCMD to the command-line before executing it. For example,
if you type "SET DIRCMD=/2" and then type "DIR *.*", IMPOSTER will execute "DIR *.* /2". If you type "DIR *.* /w", however, IMPOSTER will not use DIRCMD since you specified a switch parameter (/w).

ENTAB

Format: ENTAB [<tab size>]

ENTAB is an IMPOSTER filter, which will search the input stream for sequences of blank characters and replace them with tab characters.

The tab size tells ENTAB how many characters are between tab stops. The default tab size is 8 characters if the tab size is not specified.

Like all IMPOSTER filters, ENTAB will accept input only from a redirected file or a pipe.

ERASE/DEL

Format: ERASE <path>

Deletes file(s) specified by the path.

DUMP

Format: DUMP <filespec> [<output file>] [/P]

DUMP is a simple utility for viewing any file stored under DOS. It presents the file in hex format to the file specified.

If no file is given, DUMP assumes the console is the output device.

If the /P switch is specified, the output will be paused after each screenful. The /P switch is valid only when the output device is the console. The /P switch is invalid when outputting to a file or to any other device.

EXIT

Format: EXIT [WINDOWS]

Exits IMPOSTER and returns you to Windows. Note that it is not necessary to exit IMPOSTER to use other Windows applications, due to the multitasking nature of Windows.

If the WINDOWS parameter is specified, you will exit Windows and return to DOS. Note that all currently running Windows apps must also agree to return to DOS as well.

If IMPOSTER is the shell, typing EXIT at your first instance of IMPOSTER will also exit Windows.

FIND

Format: FIND [/V] [/C] [/N] [/I] ["] <searchstring> ["]

FIND is an IMPOSTER filter, which will search the input stream for the specified search string. If the search string is only one word, it can be specified without double quotes; otherwise, double quotes will be needed before and after the search string (i.e. 'DIR | FIND "Directory of"').

If the /V switch is specified, only lines that do not match the search string will be displayed.

If /N is specified, relative line numbers will also be

displayed.

The /C switch tells FIND to only display a count of all lines that matched. The /C switch overrides the /N switch. If /C and /V are both specified, FIND will display a count of all lines that did not match.

By default, the search is case-sensitive. The /I switch tells FIND to ignore case while performing the search.

Like all IMPOSTER filters, FIND will accept input only from a redirected file or a pipe.

FF/WHEREIS

Format: FF [<filename/pattern>] [/A] [/P]

Searches for all files matching the pattern on the current drive. Will display all matches. If no pattern is specified, all files are displayed.

If the /A switch is specified, it will search all drives for the pattern.

If the /P switch is specified, screen output will pause each time the screen fills.

FS

Format: FS [<path>] [/S] [/T] [/P]

Reports the file sizes for all files matching the path. If no path is specified, the current directory is used.

If the /T switch is specified, only the total of the file sizes is reported. If /S is specified, subdirectories will also be searched. If /P is specified, the screen will pause as it fills.

HELP

Format: HELP [<command>]

Brings up the IMPOSTER help system. If a parameter is specified, IMPOSTER will bring up command-specific help for that parameter.

For example, if you type "HELP DIR", IMPOSTER will load the help system and immediately position you to the DIR command help screen.

Note that the same effect can be gained by using the /? parameter with any internal command (e.g. "DIR /?").

KEYSTUFF

Format: KEYSTUFF <char> | #<num> [<char> | #<num>] ...

Stuffs characters into the keyboard buffer to simulate a user typing on the keyboard.

Characters are interpreted literally on the command line, unless preceded by a '#' character, in which case the number following the '#' is treated as an ASCII value. All characters must be separated by spaces.

E.g. "KEYSTUFF Y #13" will stuff a 'Y' character and a carriage-return into the keyboard buffer.

LABEL

Format: LABEL [drive] [label]

Allows you to update the volume label on the specified drive. If no drive is specified, the current drive is used.

If no label is specified, the following prompt is displayed:

```
Volume in drive C: is XXXXXXXXXXX
Type a volume label of up
to 11 characters or press <Enter>
for no volume label update:
```

If the user presses <Enter> without entering a label:

Delete current volume label (Y/N)?

If the user says yes, the volume label is deleted.

MAX

Format: MAX <programname> [<parameters> ...]

Executes the specified program, maximizing it. This command is essentially the same as just typing the programname with parameters, except that the application starts out maximized.

MEM

Format: MEM

Displays free global memory and system resources.

MIN

Format: MIN <programname> [<parameters> ...]

Executes the specified program, minimizing it. This command is essentially the same as just typing the programname with parameters, except that the application starts out minimized.

MKDIR/MD

Format: MD <path>

Creates the specified directory.

MORE

Format: MORE

MORE is an IMPOSTER filter which will automatically prompt the user if they wish to continue after the output has filled the virtual screen.

Like all IMPOSTER filters, MORE will accept input only from a redirected file or a pipe.

MOVE/RED

Format: RED <path1> <path2> [/V] [/!]

Moves file(s) from path1 to path2. Path2 must be only a directory specification. Note that if the paths are on the same drive, the files will be moved simply by renaming; otherwise, they are copied and then deleted.

If the /V switch is specified, files will be verified byte for byte as they are copied from one drive to another. If the paths are on the same drive, the /V switch has no effect.

Note that MOVE will by default ask permission before overwriting existing files. If the /! switch is specified, MOVE will overwrite existing files without asking.

NUKE

Format: NUKE <path> [/Z] [/Q] [/N] [/!]

Completely erases the specified subdirectory. Note that NUKE deletes subdirectories, not individual files. Unless the /! parameter is specified NUKE, will ask for permission before removing the specified directory tree. Also note that it will only ask once.

For example, "NUKE C:\ /!" could be very destructive if you didn't intend to delete all files and directories on drive C:.

If you specify the /Z option, NUKE will physically overwrite and rename deleted files with binary zeros to ensure files cannot be recovered. This option takes a long time, however.

If you specify /Q, NUKE will not echo any information to the screen while removing the subdirectory.

Last, if you specify /N, NUKE will not remove subdirectories within the directory you specify; i.e. it will only delete files from the first directory level.

PATH

Format: PATH [<path>]

Displays the current path if no parameters are specified. Otherwise, sets the path to the specified parameter.

Note that the same thing can be accomplished by using the SET command on the PATH environment variable.

Last, IMPOSTER by default always additionally searches the %WINDIR% directory and the SYSTEM subdirectory underneath it, since Program Manager does the same.

PROMPT

Format: PROMPT [[text] [$character] ...]

Sets the DOS prompt to use. By default, IMPOSTER uses the PG prompt.

The following special characters are supported:

$q	The = character.
$$	The $ character.
$t	Current time.
$d	Current date.
$p	Current drive/directory.
$v	DOS version.
$g	The > character.
$l	The < character.
$b	The | character.
$_	RETURN-LINEFEED.
$-	The blank character.

ANSI escape sequences are not supported. Other characters may be included in the prompt line and will be displayed literally.

Note that the same thing can be accomplished by using the SET command against the PROMPT environment variable.

RENAME/REN

Format: RENAME <path> <filename>

Renames the files specified by path to the filename mask specified by filename. If wildcards are used, the positions of the corresponding characters are literal.

RENDIR

Format: RENDIR <path> <filename>

Renames the subdirectories specified by path to the filename mask specified by filename. If wildcards are used, the positions of the corresponding characters are literal.

RMDIR/RD

Format: RD <path>

Removes the specified directory.

SET

Format: SET [varname=[text]]

Displays the current DOS environment, if no parameters are specified.

If parameters are specified, they must take the form of "varname=text ...", with no space before or after the equals sign. The environment variable will be updated or

added as appropriate. If no text is specified after the equals sign, the environment variable will be removed, unless the variable is the PATH or COMSPEC environment variable.

Note that only the local environment to the instance of IMPOSTER you are running is updated. The environment given to other Windows apps is unchanged.

SORT

Format: SORT [/R] [/+<offset>]

SORT is an IMPOSTER filter, which will sort the input stream using the specified parameters and write it to standard output.

If the /R switch is specified, the sort will be reversed (i.e. sorted in descending order).

The /+ tell SORT which column to begin sorting on. For example, specifying /+14 tells SORT to only consider columns 14-EOL when performing the sort. Multiple sort keys are not supported.

Like all IMPOSTER filters, SORT will accept input only from a redirected file or a pipe.

TIME

Format: TIME [hh:mm:ss]

Updates the current DOS time. If no time is specified, the following prompt appears:

```
Current time is 13:05:00.
Enter new time:
```

If the user hits enter at this prompt, the time is not changed.

TOUCH

Format: TOUCH <path>

Resets the DOS date/time on all files described by the path to the current date and time.

TREE

Format: TREE [<directory>] [/P]

Displays all subdirectories underneath the specified directory. If no directory is specified, the root of the current drive is assumed.

If /P is specified, the display will pause each time the screen fills.

TS

Format: TS <path> <searchstring> [/CS] [/LOG] [/S]

Searches all files in the path for the searchstring.

If the /CS switch is specified, the search is case-sensitive, otherwise it is not. If /LOG is specified, the program will run non-interactive and will produce more batch-oriented output. If /LOG is not specified, the program will pause after every match is found and ask if you want to continue the search.

If /S is specified, TS will search all files in all subdirectories as well.

TYPE

Format: TYPE <filespec> [/P]

Displays the contents of the selected file to the console.

If /P is specified, the screen will be paused after the screen

fills.

VER

Format: VER

Displays the IMPOSTER, DOS and Windows version numbers.

VERIFY

Format: VERIFY

Displays the current status of the DOS VERIFY flag. Does not allow you to change the setting.

VOL

Format: VOL [drive:]

Displays the volume label on the specified drive. If no drive is specified, the current drive is used.

WINDOW

Format: WINDOW [MAXIMIZE | MINIMIZE | RESTORE]
WINDOW <x> <y> <width> <height>

Adjusts the size of the IMPOSTER window. If MAXIMIZE is specified, the IMPOSTER window will be maximized. If MINIMIZE is specified, IMPOSTER will become an icon. If RESTORE is specified, IMPOSTER will be "restored" to its regular size if it was previously maximized or minimized.

If the alternate format is used, the IMPOSTER window will be placed with the upper-left corner at x:y and with the specified width and height. All values must be numeric and are expressed in physical device coordinates (i.e. pixels).

XCOPY

Format: XCOPY <path1> <path2> [/S] [/E] [/A] [/M] [/V] [/T]

Copies file(s) from path1 to path2, using the specified switches. All copies are binary mode.

The switches supported are:

/S Copy all subdirectories except empty subdirectories.
/E Copy empty subdirectories; used only with /S.
/A Copies only files that have the archive bit set.
/M Clears the archive bit after each file is copied.
/V Compares source and destination files byte for byte after each copy.
/T Forces copied files to have the same attributes as the source files.

External Applications

If a command is entered that is not on the list of supported internal commands, IMPOSTER assumes you wish to run another application by that name. It then searches the current directory and the DOS path for the application.

Unlike the DOS command interpreter, IMPOSTER will look for any file with the extension listed by the "Programs" keyword in the [Windows] section of the WIN.INI file. The extensions must have no period and be separated by spaces.

By default, Windows defines these extensions to be .COM, .EXE, .BAT and .PIF:

```
[Windows]
Programs=com exe bat pif
```

IMPOSTER will look for a matching application by the order specified in this WIN.INI parameter. So, if you were to type "NOTEPAD" at the IMPOSTER command prompt, IMPOSTER would search the current directory and the DOS

path for NOTEPAD.COM; if it could not find one, it would then search for NOTEPAD.EXE, probably finding it in the WINDOWS directory.

Note that if you run a DOS command, IMPOSTER will spawn a copy of DOS to run the command. So, for example, if you type "123" at the command prompt, IMPOSTER will load DOS and run Lotus 1-2-3 (assuming you have 1-2-3 and it is locatable on the path). Note that since .EXE comes before .PIF in the example above, IMPOSTER will run 123.EXE before 123.PIF. It is for this reason that IMPOSTER juggles the order of the PROGRAMS= line to look like this:

```
[Windows]
Programs=pif exe com bat
```

In this way, IMPOSTER will run corresponding .PIF files instead of corresponding .EXEs.

Last, IMPOSTER makes full use of the [Extensions] section in WIN.INI. This is a section usually used by the Program Manager to allow relationships between types of files and the applications that manipulate them. IMPOSTER uses the same approach.

The extensions section typically looks something like this:

```
[Extensions]
txt=notepad.exe ^.txt
doc=notepad.exe ^.doc
ini=notepad.exe ^.ini
```

These definitions all run NOTEPAD.EXE, if you enter a filename that has a .TXT, .DOC or .INI extension. So it is possible, for example, to type the following at the IMPOSTER command prompt:

```
C:\WINDOWS\WIN.INI
```

IMPOSTER will notice that the file has a .INI extension, so it will scan the [Extensions] section looking for an "INI=" line. If it did not find one, it would indicate the file cannot be run.

If it does find one, however, it notices that the .INI extension is associated with the NOTEPAD.EXE application. IMPOSTER then executes NOTEPAD.EXE and passes C:\WINDOWS\WIN.INI as a parameter, exactly as if you had typed in:

```
NOTEPAD C:\WINDOWS\WIN.INI
```

If you wish to add new application/file extension relationships, just edit the WIN.INI and enter them as you see the others. For example, if your version of Lotus 1-2-3 saves files with a .WK1 extension, you could establish a relationship as follows:

```
[Extensions]
WK1=123.EXE ^.WK1
```

Be careful in these definitions, however. Sometimes file extensions are not unique to the application that generated the file. For example, the .DOC extension is used very often by very different word processing packages with incompatible file structures; likewise, many people use .DOC as the file extension even for simple ASCII files.

Batch Files

IMPOSTER also fully supports standard DOS batch files. If you enter the name of a batch file at the IMPOSTER command prompt, IMPOSTER will execute the batch file the same way DOS will.

If the batch file is found, all commands in the batch file are executed in sequence. All standard batch file conventions are supported, although some of the more esoteric quirks of the DOS COMMAND.COM interpreter may not function quite the same way.

Another difference is that, unlike DOS, IMPOSTER may not wait until a program is done executing if it is an external application. For example, if you have a batch file called WORDPROC.BAT:

```
@echo off
REM This batch file loads all
REM my favorite word processors.

ECHO Running Windows Notepad...
NOTEPAD

ECHO Running Brief...
B

ECHO Running Windows Write...
WRITE

ECHO Running Word for Windows...
WINWORD
```

Since Notepad, Write and Word for Windows are all Windows applications, IMPOSTER will only pause long enough for the application to load before continuing on to the next line in the batch file. The result? Notepad, Write and Word for Windows will all be running simultaneously.

Now what about Brief? Brief is a DOS-based editor, so will IMPOSTER wait until the user exits it before proceeding on with Write and WINWORD? If you run Windows in 386 Enhanced Mode, the answer is often no. IMPOSTER will run Brief in a window if you have a properly-designed .PIF file defined for it. Just make sure the .PIF extension appears before all others in the Programs= line of the [Windows] section in WIN.INI. Once Brief is loaded in the Window, IMPOSTER continues on to the other lines in the batch file.

If you run the B.EXE program with no .PIF file or if you are running Windows in Standard or Real Mode, IMPOSTER will run Brief full screen, and yes, IMPOSTER will wait until the user exits Brief before continuing on with Write and Word for Windows.

Arguments are also supported just as they are in DOS. The first command-line parameter is referred to as %1, the second as %2, and so on. The %0 parameter is also available and always refers to the name of the batch file that is running.

Environment variables are also expanded as they are in DOS. Take as an example:

```
Echo Current path is "%PATH%"
```

IMPOSTER will replace %PATH% with the contents of the PATH environment variable. This is consistent with DOS.

The @ prefix also works with IMPOSTER as it does with COMMAND.COM. If you prefix any line with the @ sign, that line will not be echoed to the console, even if command echoing is on. This works with any command, not just the ECHO command!

Note that IMPOSTER allows two types of batch files--those that have a .BAT extension and those that have a .IPS extension. Regular .BAT batch files function almost exactly as they do under DOS, as described above. The .IPS batch files, however, are provided for future expansion when IMPOSTER allows more powerful but nonstandard extensions for the Windows environment. In the meantime, .IPS and .BAT files are interchangeable.

One important caveat, however: sometimes you may wish to execute a batch file whenever you startup Windows. To do this, you must have the .IPS extension associated with IMPOSTER (look in the [Extensions] section and see if the line "IPS=IMPOSTER.EXE ^.IPS" is there). Once this association is made, you can simply specify "LOAD=ANYBATCH.IPS" or

"RUN=ANYBATCH.IPS" in your WIN.INI and the Windows shell will run IMPOSTER, execute the ANYBATCH.IPS batch file and then close IMPOSTER. This is a nice way to run multiple Windows applications, with parameters, whenever you start Windows. Unfortunately, you cannot do the same with .BAT files--the Program Manager will simply assume .BAT files were meant for COMMAND.COM and refuse to run IMPOSTER.

Batch File Commands

Some commands are useful only within a batch file. IMPOSTER supports all of the standard DOS batch file commands, as well as some enhancements:

ASK

Format: ASK "text" <keylist>

ASK will display the text specified in quotes to the console and wait for the user to press a key in the keylist. The DOS ERRORLEVEL will be set to the key number, as listed left to right.

For example:

```
ASK "Continue processing (y/n): " yn
```

If the user hits the Y key (whether shifted or not), ASK will return DOS ERRORLEVEL 1. If the user hits the N key, ASK will return DOS ERRORLEVEL 2.

CALL

Format: CALL <batchfile> [arguments...]

Calls the specified batch file. Control returns to the calling batch file after the sub-batch file is executed.

This differs from simply declaring the batch file name in that control does not return to the calling batch file if CALL is not used. The arguments used are passed to the batch file just as if the batch file had been typed at the command prompt with arguments.

CTTY

Format: CTTY CON | NUL | <filespec>

Changes the output device to the specified device or file. If CTTY CON is specified, all output will go to the console device (the IMPOSTER window). If CTTY NUL is specified, no output is displayed.

If something besides CON or NUL is specified, CTTY attempts to send all output to a file by that name.

Unlike DOS, if you fail to reset to CTTY CON at the end of a batch file, IMPOSTER will reset it for you.

ECHO

Format: ECHO ON | ECHO OFF | ECHO <message>

Displays a message to the screen, turns off command echoing or turns on command echoing.

If ECHO ON is specified, command echoing will be switched on; likewise, ECHO OFF turns off command echoing. If the ECHO: variant is used, no message is allowed and a blank line is displayed on the console.

Remember that the @ prefix can be very useful with ECHO, particularly as the first command in a batch file (e.g. "@ECHO OFF").

FOR

Format: FOR <%%c> IN (<set>) DO <command <%%c>>

Executes the command for all files found in the set. The set can be a comma or space delimited list of paths or can be a single wildcarded specification, but cannot use both. Note that the set must be bounded by parentheses or a syntax error will occur. Command-line parameters may be used in the list if desired.

The %%c parameter is simply a placeholder that tells IMPOSTER where to substitute the argument in the command. This placeholder must be exactly three characters long and the last character must be alphabetic (e.g. %%A through %%Z or %%a through %%z).

IF

Format: IF [NOT] ERRORLEVEL <number> <command>
IF [NOT] <string1> == <string2> <command>
IF [NOT] EXIST <path> <command>

Evaluates the expression and, if true, executes the specified command.

GOTO

Format: GOTO <label>

Jumps forward or backward to the specified label.

:<label>

Format: :<labelname>

Declares a label within a batch file. Otherwise is not executed.

PAUSE

Format: PAUSE [message]

Pauses execution of the batch file. Either displays the specified message or displays:

Strike a key when ready...

If there are command-line parameters, PAUSE will simply display them. For example, "PAUSE Hey, hit a key bozo!"

will result in the following prompt for the user:

Hey, hit a key bozo!

Any additional spaces between words in the message are ignored.

REM

Format: REM <remarks>

Ignored by IMPOSTER. Allows remarks to be placed in a batch file.

SHIFT

Format: SHIFT

The SHIFT command simply allows more than ten parameters to be accessed by your batch files (%0 to %9). When you specify SHIFT, all parameters are shifted down by one, so that %1 becomes %0, %2 becomes %1, etc.

The previous contents of %0 are lost and there is no backward shift command to retrieve it, even if you have less than ten parameters.

Appendix A - Common Questions and Answers

With each release of IMPOSTER, I've found that the same questions are often asked relating to the functionality of IMPOSTER. Here are some of these questions and some (hopefully) helpful answers:

Q: How can I set IMPOSTER up to always start maximized?
A: Simply set the startup command to be "WINDOW MAXIMIZE" in your preferences, or use the "Run Minimized" Properties option from the Windows 3.1 Program Manager.

Q: What if I have multiple commands I want to execute at startup?
A: It may generally be a good idea for most users to set up a batch file to always execute upon startup of IMPOSTER. I like to call mine STARTUP.IPS, and I just have it hidden away somewhere on the path. In STARTUP.IPS, I set my prompt, set IMPOSTER's initial window position and size, and set the path to a special path I only want to use from IMPOSTER. For example:

```
@echo off
prompt $d $t$_$p$g
set path=c:\windows;c:\winbin;c:\bin;c:\tools;c:\bat
window 10 10 1000 1000
```

Then I just set the startup command to be "STARTUP.IPS". IMPOSTER does the rest!

Note you can accomplish the same thing by using multiple commands in the startup
command as well.

Q: I want to user IMPOSTER primarily to execute batch files but want to be able to trigger the batch files from icons in Program Manager. Any ideas?
A: Yes! IMPOSTER takes command-line parameters too, just like any other DOS application. If you specify parameters, IMPOSTER will execute those parameters and return to Windows after execution is completed. This is useful for running batch files. For example, you can create a Program Manager icon that calls "IMPOSTER BKUP.BAT" to XCOPY your hard disk over to a network drive. Then whenever you doubleclick on this icon, IMPOSTER will load, execute the batch file, and then return.

This also has some interesting uses within batch files. What if you want to be able to spawn multiple batch processes to copy files all over

simultaneously? Assume you have a batch file called WILDCOPY.IPS:

```
@echo off
min imposter copybin
min imposter copywin
min imposter copybat
```

And assume COPYBIN.BAT, COPYWIN.BAT and COPYBAT.BAT all copy different directories on your hard disk to various directories on a network. When you type WILDCOPY at the IMPOSTER command-line, IMPOSTER will load itself again three times and *simultaneously* perform the copies you specified. And since you loaded IMPOSTER with the MIN command, all three background processes will run as icons to boot!

Another interesting thing you can do using this technique is to create a batch file called "SPAWN.IPS" or "FORK.IPS":

```
@min imposter %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Such a batch file would allow you to do things like type "SPAWN XCOPY *.GIF M:\BKUP", which would run a second copy of IMPOSTER as an icon and perform the specified copy in the background in that second process. It is especially useful now that I/O redirection and piping is supported, which allows you to do things like "SPAWN FF *.GIF /A > SEARCH.TXT" and see the results of the search later in the SEARCH.TXT file. Imagine the possibilities!

Q: I'm confused about the difference between .IPS and .BAT batch files. What gives?

A: The answer is that there is no difference as far as IMPOSTER is concerned, for the time being. The real significance is that COMMAND.COM won't find .IPS batch files, so you may want to put IMPOSTER-specific scripts in .IPS batch files, while more common batch files could go in .BAT batch files. It's really up to you, but is provided as a way to prevent errors when running your batch files from the real COMMAND.COM.

At some point, I plan to add Windows-specific commands to make batch processing more powerful. When that occurs, IMPOSTER will limit these extended commands to .IPS batch files only, further protecting you from confusion when you are running the real COMMAND.COM.

Q: I want to run a batch file from my LOAD= or RUN= line in WIN.INI, but

I'm not able to specify a parameter alongside IMPOSTER (due to the way Windows handles these lines). How can I do this?

A: Since IMPOSTER associates the .IPS extension with IMPOSTER, you can simply do this:

LOAD=ANYBATCH.IPS DRWATSON.EXE SAVER.EXE

Whenever you start Windows, IMPOSTER will load and execute the ANYBATCH.IPS script. If IMPOSTER is not your Windows shell, IMPOSTER will then exit.

This is a useful technique for executing one set of batch commands when you start Windows and another when you start any instance of IMPOSTER (through the startup command).

Note also that if you run IMPOSTER as the shell, you can run regular .BAT files in the same way.

Q: The IF command mentions the ERRORLEVEL, which DOS uses as a return code from programs or internal commands. How does this work with IMPOSTER?

A: The ERRORLEVEL is fully emulated in IMPOSTER for all commands IMPOSTER executes. Generally, ERRORLEVEL 0 means your command was successful. ERRORLEVEL 1 means a minor error (like a syntax error) occurred, and ERRORLEVEL 2 means a bigger error occurred.

Unfortunately, there's no real way to return an ERRORLEVEL from a program you launch from IMPOSTER, so the ERRORLEVEL only indicates if the program was successfully launched in these cases.

Q: The DOS path just isn't big enough to point to all my DOS application directories and my Windows application directories at the same time! Is there anything I can do?

A: Since IMPOSTER supports environment variables, you can now define your own local path for IMPOSTER to use. An excellent place to define this path is in your startup command (or in STARTUP.IPS, if you use that convention). Additionally, IMPOSTER allows much longer environment variables than DOS and isn't constrained by artificial environment size limitations. Note that this path is local only to the current instance of IMPOSTER and will not be visible to DOS boxes or other Windows apps.

Q: Does IMPOSTER use _DEFAULT.PIF or what when executing DOS commands?

A: This is dependent upon whether you have a .PIF file already defined for

the command. If you do, IMPOSTER will use that .PIF file without consulting _DEFAULT.PIF. If you do not have a .PIF file, however, IMPOSTER will use _DEFAULT.PIF to set up the DOS environment for the command.

Q: I like to run DOS commands from IMPOSTER but sometimes the DOS window goes away too quickly for me to see the results of the command. For example, if I type PKZIP -v ANYFILE.ZIP, the window destroys itself before I can see the end of the listing. Is there a way for IMPOSTER to prevent this?

A: Actually, there is no way (I think) for IMPOSTER to prevent this but Windows fortunately provides a solution. If you create a .PIF file for each DOS command you use, you're already halfway there. Just ensure that the "Close Window on Exit" checkbox is NOT checked for programs that have this problem and Windows will wait for you to close the window manually before destroying the window.

Q: IMPOSTER is just such a great program and is such a wonderful contribution to Windows users everywhere that I am just compelled to send money to ease my conscience. What should I do?

A: Well, I am not distributing IMPOSTER to make money in any way. I want IMPOSTER to be a publicly-available program that everyone can enjoy without having to spend any money on it. I can remember the good old days of public domain software (back when DOS was king) and there really was a lot of good stuff available just for the asking. I'm hoping a similar phenomenon occurs with Windows.

If, however, you feel you really must make a contribution as thanks for the huge amount of time I've invested in creating IMPOSTER, then I have two alternatives for you:

- Donate a few bucks to your favorite charity. That should really make you feel good and helps your community as well. If you let me know, it will also make me feel good.

- Donate a few bucks to Andy Walding of the Digital X-Connect BBS, who has graciously allowed me to use his BBS as the official distribution site. He really has a great board going and can always use some generous contributions to offset his cost in providing the BBS to the public. Just log on to the X-Connect as described in the technical support section and follow the BBS's instructions for becoming a contributor.

Appendix B - Planned Features

Just so everyone knows what the "plan" is, I thought I would document IMPOSTER's future direction. If you have any questions, suggestions or concerns about this schedule, please let me know as described in the technical support section.

Planned Features

- 4DOS/NDOS command emulation. This is the most highly-requested feature, so definitely look for IMPOSTER to emulate many of the commands in these products.
- Windows-specific commands. IMPOSTER is probably an ideal platform to add a host of Windows commands to make managing Windows easier and accessible from a batch file. For example, the ability to kill processes, the ability to create messages boxes, etc. would all be useful features.
- Windows 3.1 support. Currently IMPOSTER is Windows 3.x compatible, meaning that it will run under either Windows 3.0 or 3.1 without modification. Look for features specifically for Windows 3.1 users, such as improved font support (potentially TrueType support), some support for drag-and-drop, potential support for OLE, etc.
- General command improvement. Many of IMPOSTER's commands can use new features, with too many potential improvements to list here. Many other DOS commands should be added, such as FC, COMP, DISKCOPY, etc.

Potential Features (Still undergoing R&D)

- DOS program emulation. With this enhancement, I would envision IMPOSTER having the ability to actually execute a DOS program within the IMPOSTER window. Actually, IMPOSTER wouldn't actually be executing the code, but it would look that way! This may or may not be practicable, but I'll give it a shot.
- OS/2 2.0 support. I've had several people report that IMPOSTER runs under OS/2 2.0 but isn't very useful since you can't spawn DOS apps from it. Depending on how OS/2 fares over the next several months, I may decide to add full OS/2 2.0 support at some point.
- Windows NT support. I've had one user report that IMPOSTER works under an alpha version of NT, but there are probably a lot of things that IMPOSTER should do given NT's new functionality. Look for NT support when NT becomes more of a reality.
- DOS support. Here's a weird idea: have IMPOSTER.EXE work from DOS as well as from Windows, providing the inverse support for the DOS

environment. I'm still looking at this, since at this point I don't think there would be a lot of benefit to the user (aside from the additional commands IMPOSTER supports over DOS).

Appendix C - Revision History

Version 1.01 - First production update:

- Added improved font support.
- IMPOSTER now remembers the insert/overstrike mode between sessions.
- Fixed a problem with the sort filter writing to the console.
- IMPOSTER now exits windows if it is the shell and you kill it via TASKMAN.
- Fixed a stupid bug with the piping/redirection feature. Turns out one of my last-minute "fixes" to another problem broke it!
- Fixed insert mode; this was also a victim of the previously mentioned "last minute fix".
- Removed the annoying beep upon IMPOSTER exit--this was a piece of debugging code I basically forgot to take out....sigh...

Version 1.00 - Final production release:

- Added support for I/O redirection and piping.
- Added support for filters.
- Added hotkey support.
- Added support for multiple commands per command-line.
- IMPOSTER now remembers its last window placement and size! Finally!
- Added a preference for the default state of launched applications (minimized, maximized, normal or hidden).
- Made MOVE ask permission before overwriting files; added an option to force MOVE to overwrite without asking.
- Added switches to DIR to recurse subdirectories, to show only subdirectories, to show a bare listing, to use lowercase and optionally condense filenames in the output listing. Also made the directory listing look more like DOS 5.0.
- Added the TREE command.
- Added the KEYSTUFF command.
- Added the CDD command.
- Added the RENDIR command.
- Added the /S and /Q switches to the ATTR command.
- Added conventional (AM/PM) time to the TIME command.
- Enhanced CTTY to allow redirection of all console output to a file.
- Enhanced the FOR command to work with sets without comma delimiters.
- Made the PATH and PROMPT environment variables function like they do under DOS.
- Trapped "evil" DOS commands like CHKDSK, APPEND, SUBST, etc. that should not be run after Windows is started. Also acknowledged the presence of commands like CHCP, KEYB and GRAFTBL in DOS 5.0.
- Added the /T switch to XCOPY.

- Added the /P switch to FF.
- Added the /S switch to TS.
- PROMPT now behaves more like DOS 5.0 in that a null prompt defaults to the drive letter (e.g. D>).
- XCOPY now behaves more like DOS 5.0.
- Fixed a bug where COPY, XCOPY and MOVE would preserve file attributes as well as the date/time. These no longer preserve the attributes but still preserve the date/time.
- Fixed a bug in MOVE when attempting to rename while moving.
- IMPOSTER now detects if you use the mouse to click on the command-line. Previously, IMPOSTER would go spastic if you did this.
- Made the CD command behave more like DOS 5.0.
- Ensured all dialog boxes use the keyboard in a more consistent form.
- The system menus on all dialog boxes now work properly.
- IMPOSTER no longer scrolls right when you uniconize the terminal window.
- IMPOSTER now detects it is the Windows shell if you manually specify SHELL=IMPOSTER (i.e. no extension).
- Fixed a bug in the DUMP command where space characters were displayed as a '.' in the cheater block.
- Fixed a bug where external commands could not be executed from batch commands like FOR or IF.
- Fixed a terrible bug where "MOVE C:*.* ." would copy the files from C:\ to the current dir but then erase them in the current dir instead of in C:\. Dumb!
- Fixed a problem where the cursor would intermittently lose its mind when on the 24th line of the terminal.
- Fixed a bug in ATTR where you could never turn attributes off. I can't believe nobody found this one.
- Improved the error messages when the user tries to violate file attributes.
- Fixed a nasty UAE that occurred when IMPOSTER encountered an environment variable longer than 127 characters. IMPOSTER now fully supports environment variables up to 255 characters.
- Fixed a bug where the DUMP command always returned ERRORLEVEL 2.
- Fixed a bug where you could not set environment variables from within an IF or FOR command.
- Fixed a problem where "WIN.INI" would not associate with Notepad if the WIN.COM file was in the current directory.
- Fixed a bug (again) where trailing blanks were left out of commands like ECHO, PROMPT, etc.