

ATARI TOS DESKTOP

SURVIVAL KIT

A GUIDE TO THE DESKTOP.INF AND NEWDESK.INF FILES

by Thomas J Hopper

CONTENTS

Legalities	-	Rights, etc.
Welcome	-	An introduction of sorts
DESKTOP/NEWDESK Basics	-	Get to know the INF files
Editing the INF File	-	Basics about editing the INF files
The Keyboard and Installing Apps	-	Keyboard functions
Neat Hacks	-	Some suggestions
INF File Composition	-	Explanation of INF file structure
Annotated NEWDESK.INF file	-	A real-life example

LEGALITIES

I make no warranty as to the usefulness or accuracy of this document.

copyright © 1994, 1995, 1996 Thomas J Hopper

This file may only be distributed in its entirety and at no charge. No portion of this document may be distributed for profit without the explicit permission of the author.

WELCOME!

There are lots of things you can do with the new Atari desktops (and with the old ones!); almost as much as with

a replacement desktop like NeoDesk from Gribnif Software! Unfortunately, the manuals Atari gives us with the computers don't tell you about all of these great features. I will try to rectify this deficiency.

I have compiled as much information on the NEWDESK.INF file that I could, and thrown in some info on TOS 2.06. This file focuses on Atari's NEWDESK.INF file, and in particular on TOS 2.06, since they are what I have. If you don't have TOS 2.06, don't worry; most of the really useful info in here is applicable to *all* versions of TOS, and some info specific to TOS 4.0x is included, too. I have tried to point out where information may be specific only to certain versions of TOS. With this as a reference, you should be able to modify your DESKTOP/NEWDESK.INF file to do almost anything with your desktop!

I have learned all of this partly through trial and error, and partly through discussions on online services. A lot of the information came to me by way of the NEWDESK topic on GENie in the Atari Roundtable. GENie, by the way, is an excellent source of information and support for the Atari line of computers; having been the only official online Atari service, there are more Atarians there than almost anywhere else. I suppose I should state that I have no connection with GENie other than as a (former) satisfied user. Another excellent source of info and programs is the Internet, where Atarians from all over the world can (and do!) converse and share programs.

If you find that something is wrong, please let me know! If you see something I missed, or if you think I should include some information that isn't in here, let me know that, too. And of course, suggestions are always welcome. When contacting me, include your machine type, TOS, GEM, AES, MiNT, and MultiTOS versions (whichever apply). This will help me decipher different, unknown parameters. If you don't know some of this information or don't have MiNT or something, don't let that stop you from contacting me! But please, try to include at least the TOS version (TOS date will suffice, if necessary; just give me the copyright dates in the "Desktop Info..." menu). Also include your name and some way that I can get in touch with you, such as email address or postal address.

I can be reached at:

tjh@ic.net

DESKTOP/NEWDESK BASICS

The DESKTOP.INF and NEWDESK.INF files in Atari computers contain all the information your computer needs to configure itself. It contains all the names, positions and (for appropriate TOS versions) icon numbers for your desktop icons. It stores all the info on Installed Apps, desktop icons, and window positions (even for windows that have been closed!). If you leave a window open when saving the desktop, the location, current directory, and file mask (for those versions of TOS that support masks) are saved. It stores info on whether or not the key click and system bell are turned on, the repeat rate for key clicks, mouse double click rate and tracking rate, current resolution, keyboard shortcuts for menu items (in TOS \geq 2.0x), and lots more. In short, it stores every piece of information that you can configure from the desktop or the standard Atari Control Panel.

When you boot up your machine, the computer loads in auto programs and then reads the DESKTOP/NEWDESK.INF (hereafter referred to collectively as simply "INF") file for information on resolution, windows, etc. If you have Atari's Control Panel (or a replacement), the key repeat rate, printer and serial port configuration, and other information is also loaded from the INF file and configured.

To change the INF file, you need simply alter a setting from either the Control Panel (not the extensible control panel, however) or one of the desktop menus, and then select "Save Desktop." If you do this several times and display the INF file after each save, you will get a feel for which changes to the desktop correspond to changes in the INF file. Before doing this, of course, make a backup of your original INF file by renaming it to DESKTOP.INX. This way if you somehow "break" the INF file, you can always be sure to have a working one handy.

You can view the file directly from the desktop; the INF file is simply an ASCII file! This means that not only is everything in the file plainly visible from the desktop, but you can load it into just about any word processor, text editor, or DTP program and edit it by hand. This has the advantage of allowing you to make certain changes that the desktop doesn't allow for. If you're really familiar with the INF file, you can also make multiple changes in much less time than it would take you from the desktop. The editor, however, must be able to load and save files in ASCII format, and it must not strip trailing spaces from lines (more on this in a moment).

One important bit of information to know and understand when editing the INF file is how TOS reads and applies the information in it. Not surprisingly, the INF file acts as a filter, through which TOS sifts programs and files to apply various actions (running files, assigning icons, etc.). What **is** surprising is that these filters work in reverse order from what you see in the INF file. The **last** entry in the INF file is the **first** filter applied! The way this works is as follows:

For icon assignments, TOS reads from the end of the file to the beginning, comparing each file in a directory to each filter (icon assignment) until a match is found. When TOS finds a match, it moves on to the next file. So if you have a program SQUISHII.APP, and assign an icon to it, that assignment should go toward the end of the file, that way it will be filtered out early on. If you have an icon assignment like #I xx xx xxx @ *.* @ *below* the assignment for SQUISHII.APP, then SQUISHII.APP will be filtered out at the *.* (and assigned that icon) rather than at #I xx xx xxx @ SQUISHII.APP @ . Hence, you would never see the special icon you assigned to SQUISHII.APP. This is important to understand, for two reasons. The first is that when editing the INF file, you want to work down the file from most general to most specific; getting this order wrong will give you results other than what you wanted. The other reason is that when you make an icon assignment from the desktop, it is saved at the end of the INF file. If you were to make lots of assignments, then do a generic (*.*) icon assignment, all your files would use the generic icon (until you edited the INF file to have *.* be the last filter applied).

For program assignments, TOS reads the INF file in the same way as for icons, and applies filters in the same way. When you double click on a file (whether it's executable or not), TOS starts looking for matches from the bottom of the INF file and works its way to the top. If you were to install a program VIEWER.APP for file type *.* , this assignment would be saved at the end to the INF file, and hence be the first filter TOS saw; all files, no matter what type they were, would cause VIEWER.APP to run and they would be loaded in to it. This is true even for PRGs, APPs, TOSs, etc.; in the INF file, these are just treated as applications installed with no file type. As with icons, if you edit the INF file make sure you work from most general at the top of the INF file to most specific at the bottom.

EDITING THE INF FILE

Before you begin editing your INF file, make sure that you have a backup of your current, working version! It is possible (and fairly easy, really) to change the INF file so that the computer will become "confused" and not boot!

Probably the first thing to remember when editing your INF file is that NEWDESK.INF can't be larger than 4kb (4096 bytes), and starts causing trouble with the system around 3900 bytes. Similarly, the DESKICON.RSC can only be 64kb (65536 bytes). The newer DESKICN.RSC (the color icon resource for TOS 4.0x and up) can be larger than this, though I don't know if there is a limit. If you try to exceed the limit, you'll get "Out Of Memory" errors and lots of headaches.

On earlier versions of TOS (less than 2.0x), the limit on DESKTOP.INF is even more restrictive; it can't be larger than 1kb (1024 bytes)! However, you also don't have a lot of icon assignments eating up that space, so it's a fair trade.

One excellent way around this is to have a directory full of different INF files; a specific one for each task. If you put only those application and icon assignments necessary for a particular task, you should have no problem avoiding the size limit. To switch between INF files, then you can just install either Charles F Johnson's shareware product Desk Switch 1.1 or Klaus Pederson's public domain Load Inf as the application for file types of *.INF. That way, when you want to change to a different task, just double click on the appropriate INF file, and away you go with all the key assignments, installed apps, desktop icons, and icon assignments you want for that task.

Another way around this is to get a program like GEMRAM, which loads GEM into RAM, and the program Shell Buffer (SHBUFxxx.PRG), which lets you configure the allowed size of the INF file. Shell Buffer only works when GEM is loaded into RAM.

Both solutions should work great on any Atari, and both have their advantages and disadvantages. Give them both a try and see which works best for you.

Something else to keep in mind when editing your INF file is that TOS expects to find certain formats at certain locations in the file. Line #d, for instance, has to have a certain number of spaces in it since TOS looks roughly 124 bytes into the INF file for the next line (#Z if you have TOS >=1.04 and a program set to auto boot, or #K for the menu key equivalents). That number for the bytes, incidentally, also includes carriage returns (EOL characters). Before changing your desktop file, be sure you know *exactly* how many spaces belong in line #d; the wrong number can produce unpredictable results and fatal errors. To find out, of course, just load the file into a text editor that doesn't strip trailing spaces (Word Writer, Alice, Everest, and tons more) and start counting! Be sure to count the return at the end of the line.

Most other lines (but not all of them) in the INF file require a trailing space, so if you edit your file be sure to know which ones need a space and which ones don't.

What this all means, of course, is that if you edit your DESKTOP.INF or NEWDESK.INF file make sure you're using an editor that doesn't strip trailing spaces, and which saves files as ASCII text. And, as always, ***keep a backup of your original file!***

THE KEYBOARD AND INSTALLING APPS

With the newer TOS versions, you can open a drive into a window by pressing <Alt> and the drive letter. If you want to open a drive into the current top window, you don't have to close the window and then open the drive; just hold down <Ctrl> and press the drive letter! The window will automatically be changed to the root directory of whatever drive you selected.

With TOS versions 2.0x and up, Atari has made it possible to assign a keyboard equivalent to any of the desktop menu items. That way you can select a file and hit "I" to Get Info on it. Unfortunately, the mnemonics can get pretty complicated, since there are also items that could use "I" like "Show as Icons" and "Install Icon." "D" could be used for "Sort by Date," "Delete," and "Install Devices." The desktop appears to only allow normal characters (capital A through Z) for these assignments, which gets pretty limiting. Luckily, you can also use control-key combinations! When changing the menu assignments in the Desktop Configuration menu, just hold down the <control> key while pressing your key. Now, instead of accidentally deleting a file by pressing "D," you can assign <ctrl><delete> (or <ctrl><D>, if you prefer) to "Delete," and never have to worry about accidentally deleting a file!

Take note! If you use a control-key combo for a particular menu item, that combo will no longer be available for normal desktop usage. For instance, if you assign <ctrl><D> to "Show by Date," <ctrl><D> will no longer be used to open drive D into the currently active window, instead it will be used to delete selected files. Remember:

Alternate + (A-P) = Open the drive's directory into a window

Control + (A-P) = Open the drive's directory in the active window

You can get all of the normal Desktop key commands by pressing <help> while at the Desktop.

Not only can you assign a key combo to "Delete," you can also remove the trash icon completely, and free up space for more important icons! Just select the Trash icon and then the menu item "Remove Icon." In fact, since TOS versions 2.0x and up allow you to open drives by pressing <alternate> and the drive letter, you can remove all the drive icons, too, and replace them with programs, folders, and files! If you need to actually see what's on drive D, just press <Alt><D>.

Something users of *any* TOS version can do is install more than one document type for an application! To do this, first install the application for one of the filetypes you want. Then load the DESKTOP/NEWDESK.INF file into an ASCII editor. Make a copy of the installed application line, making sure to keep all the copies together with the original in the .INF file. It will look something like:

```
#Y FF 04 000 C:\path\GEMVIEW.GTP@ *.PI3@ @
```

Then rename the installed file type ("*.PI3") to whatever other file type you want. Make sure to keep all of these lines together. After doing this a few times, you might have the following lines in your .INF file:

```
#Y FF 04 000 C:\path\GEMVIEW.GTP@ *.PI?@ @
#Y FF 04 000 C:\path\GEMVIEW.GTP@ *.PC?@ @
#Y FF 04 000 C:\path\GEMVIEW.GTP@ *.GIF@ @
```

Editing your INF file manually like this is the only way to install one application for more than one file type, since using "Install Application" from the desktop will just overwrite any previous assignment, rather than adding to it.

Of course, instead of installing an application for a particular three character suffix, you might try editing the lines for a particular *prefix*! You could do something like:

```
#G 03 04 000 C:\UTILS\FILE_VIEW.WER\AV380.PRG@ READ*.*@ @
```

So that whenever you double clicked on a file like "READ.ME" or "README.TXT," ASCII View 3.80 would be run and the file loaded into it for viewing!

In fact, you can replace the Desktop's boring old [SHOW | PRINT | CANCEL] by installing an application for all file types. Be careful; since this works as a catch-all, you have to make sure that such an installation is the *last* line TOS checks when you double click. That means it has to be the first application line in the .INF file (remember, TOS uses the DESKTOP/NEWDESK.INF file as a sort of filter, working from the end of the .INF to the beginning. Basically, the place to install an app like this is the line just before the *.APP, *.PRG, *.TOS, etc. is defined. See the annotated INF file below for an example.

NEAT HACKS

In any version of TOS you can, of course, change the name of any desktop icon to anything you want. This is true of the Trash can, which can be renamed HAZARD or whatever you want. It's also true of any programs or folders you put on the desktop (in TOS >= 2.0x); Just look at the line of the INF file where that file/folder appears. It will look something like:

```
#X 07 01 7B FF C:\UTILS\VIEWERS\VIEWER.PRG@ VIEWER.PRG@
```

The name seen on the desktop will be "VIEWER.PRG." You can change this to "SEE IT!" just by changing the line to something like:

```
#X 07 01 7B FF C:\UTILS\VIEWERS\VIEWER.PRG@ SEE IT!@
```

Of course, as I said above, you don't really need the trash or drive icons on TOS >= 2.0x, so why not just get rid of them? You can free up tons of space on the desktop for commonly used programs, files, and folders. It also has the nice side-affect of freeing up space in your INF file, allowing you to have a few more lines of icon or application assignments. This is a great tip for people who use Desk Switch or Load Inf; you can replace the drive icons with INF file icons, allowing you easy access not to your drives but to the work you need to do.

One feature that deserves to be recognized, though it isn't a hack, is the drag and drop feature of TOS >= 2.0x. With this feature, you can put your commonly used programs on the desktops and then just drag the file you want to work on over the app (until the app is highlighted), then "drop" the file. This has the advantage of requiring fewer steps to get working, as well as not requiring you to install the application in the INF file, thereby saving a few bytes of space.

Create INF files specific to programs, put them all in a common folder, and use Desk Switch or Load Inf to switch between them. You can have a DTP.INF, WORDPROC.INF, GEMDRAW.INF, DEGAS.INF, TELECOM.INF, UNARC.INF, etc. This frees up a lot of space in any particular INF file. And with each INF, each type of file used with that task can have its own unique icon, so it's easy to distinguish file types, and you can also have all the installed application info you want! Using just one INF file really limits you in what you can do, so make more!

Reset your keyboard-equivalents for the desktop menus so they make sense! Use normal keys for one menu, and <cntrl>-key combos for the other. You can even try <shft><cntrl>-key combos. Require two keys to be pressed when deleting or formatting.

Get a resource editor (like MKRSC; this works with TOS 4.0x icons) and build your own icons, then assign them by hex number to files in your INF file.

Put a folder on the the desktop and save the INF file. Then edit the INF file so that the line containing that folder begins with #X instead of #V. Now when you double click on the folder, the default document viewer is run, and the contents of the folder are loaded in a batch mode. This is an excellent way of viewing new pictures or text files! You can even use masks to control what sort of files load in. Of course, your viewer has to be capable of processing batch jobs.

Make yourself a blank icon (no mask, nothin'). You can now assign short notes to the name of the icon, such as Fkey assignments, short reminders, and whatnot. If you have TOS >= 2.0x, you can put these "sticky notes" on the desktop, too. They are easily editable from the desktop.

INF FILE COMPOSITION

Below is a (fairly) complete explanation of the lines in the DESKTOP.INF or NEWDESK.INF files. Some parameters only apply to specific versions of TOS. Compare what's below to what is already in your .INF file, and if it's not already there, *don't add it!* TOS expects certain lines to have a particular format, and adding to these lines can confuse TOS and give you headaches. On the other hand, deleting things can have the same effect, so don't do that either.

Where a description is given by "bits," the following procedure will produce the needed hexadecimal number:

start with the high bit (e.g. bit "7"), and begin writing down the desired configuration as a binary number (1's and 0's) from left to right. Unlisted bits should not be changed. When the number is complete, convert it to hex. Any decent calculator will do this without a fuss. For example: For the "Other configuration parameters" (see below), I want bit 4 set to "filename," so I write a 0; bit 3 I want set to "top window", so to the right of the zero I write another 0; then bit 2 I set "size to fit" on, so I write a 1 to the right of bit 3; bit 1 is not listed so I write a 1 (since that's what it was originally); finally bit 0 I want set to "sort on," so I write a 0. The resulting number is "00110." I set my calculator to "bin" and plug this number in, then convert it to hex by changing the calculator mode to "hex". The resulting number, which goes in the fourth column of line #E is 06.

If you aren't familiar with doing this, I suggest you try it out a couple times and compare it with what's already in your desktop file. Don't try changing anything until you are confident you can get it right! Wrong values can do unpredictable things to your system, including causing it to crash!

I have been unable to determine the usage of some bits. In these cases, I list the bit with a question mark and give the current value in my NEWDESK.INF file. Yours may be different, so check it and use whatever your INF file uses. To check it, convert the current hex value to binary: the rightmost number is bit 0, the digit to the left of that is bit 1, and so on. Converting hex 06 to bin is 110: bit 0 = 0; bit 1 = 1, bit 2 = 1, bit 3 = 0 (not shown), and bit 4 = 0 (also not shown).

And remember, if you find you've made a mistake, just delete the broken .INF file and re-load the old version that you backed up.

All "@" symbols must be followed by a space.

Serial communication settings.

Special thanks to Bill Hallman for supplying the communications settings.

```
#a000000
digit  setting                possible values
  1    duplex                  0 = full    1 = half
  2    baud                    (see below)
  3    parity                  0 = none    1 = odd     2 = even
  4    number of bits          0 = 8       1 = 7       2 = 6       3 = 5
  5    flow control            0 = none    1 = xon/xoff 2 = rts/cts 3 = 1 & 2
  6    strip bit               0 = yes     1 = no
(digits measured left to right: i.e. #a123456)
```

Baud settings:			
setting	baud	setting	baud
----	----	----	----
0	9600	8	1800
1	4800	9	600
2	1200	?	50
3	300	>	75
4	19200	=	110
5	3600	<	134
6	2400	;	150
7	2000	:	200

Printer Settings

```
#b000000
digit \ setting      0              1
1          dot matrix    daisy wheel
2          black & white  color
3          1280 dpl      960 dpl
4          draft         final
5          parallel      serial
6          continuous    single sheet
```

Keyboard equivalents

(available on TOS >= 2.0x)

```
#K xx xx xx xx ... @
next 28 columns = hexadecimal value of kbd equiv. (00 = none)
next 2 columns = 00
next column = Video prefs. key (not on all TOS versions)
last column = @
```

Video settings

(some features not available on some versions of TOS)

```
#E PR BR xx OP LD CM xx xx xx...
```

PR = Preferences (Confirm file copy, delete, and overwrite; display options)

bit 7: show files as
 1 = show as text 0 = show as icons

bits 6,5: sort
 00 = by name 01 = by date
 10 = by size 11 = by type

bit 4: confirm deletes
 1 = yes 0 = no

bit 3: confirm copy
 1 = yes 0 = no

bit 2: ?
 0

would install the program VIEWER.TTP as the default document displayer (this then supercedes the desktop [SHOW | PRINT | CANCEL]). This line also sets VIEWER.TTP to run when F9 is pressed. There are no parameters passed to this program.

Default window icons-

Example: #G 03 03 000 @ *.PRG@ @

First column = type

- #I = specific file or file type *
- #N = non-specific file *
- #D = folder
- #G = Gem app
- #Y = GTP app
- #P = TTP app
- #F = Tos app

second column = default icon # for #G, #F, #Y, #P New default icon can be installed for app type by changing this #

third column = default icon # for #N, #D New icon can be installed for non-prg files by changing this #

Both the second and third column should contain the same hexadecimal number.

fourth column - function key assignment. Leave at 000 (no assingment)

fifth column - @ filename.ext@ @ wildcard

* #I assigns an icon to the specified file or file type. It does not assign an action. #N assigns an icon to the specified file or file type and specifies it as a non-executable file. If the line

```
#N 0B 0B 000 @ *.*@ @
```

appears below the default lines for APP, PRG, etc. (i.e.

```
#G 03 FF 000 *.APP@ @ @
#G 03 FF 000 *.PRG@ @ @ etc.),
```

programs will not execute when double-clicked on.

Desktop Icons-

Example: #X 08 02 18 FF C:\UTILS\VIEWER\VIEWER.PRG@ SEE IT!@

first column = type

- #M = Storage Device
- #T = Trash
- #O = Printer
- #X = Application
- #V = Directory

second column = horizontal position in icon widths

third column = vertical position in icon widths

fourth column = Hex value of Icon number in DESKTOP.RSC

fifth column = FF (function unknown)

sixth column = Device ID (A-P) or space

Finally: full_path_and_filename@ icon label@ (for a file or folder)
Device name@ @ (for a device)

Annotated NEWDESK.INF File:

#a000000

Stores serial communication info.

#b000000

Printer Configuration

#c7770007000600070055200505552220770557075055507703111103

Color palette settings, mouse double-click response, key-click, bell sound, key delay and key repeat rate.

#d

Reserved. Needs to have roughly forty-seven spaces since GEM looks roughly 124 bytes into the file for the next line.

#Z 01 F:\RUNPROG\PROGRAM.PRG@

Program to Auto-run at bootup. Does not apply to TOS earlier than 1.04.

#K 4F 49 53 1F 46 20 43 00 41 4D 00 09 14 0E 04 13 05 0F 00 00 00 01
00 12 00 52 00 44 00 00 @

Keyboard assignments for desktop menu items. (should all be on one line)

#E 50 13 00 06

Default screen resolution; window sorting method; show as icon or text; confirm copies, deletes, etc. Second digit of the second byte is for screen resolution: 13 = blitter on, ST High rez.

#Q 41 00 43 40 43 40

#W 00 00 02 07 4B 11 00 F:\RUNPROG*.PRG@

#W 00 00 16 02 35 0B 00 @

#W 00 00 2E 0E 22 0A 00 @

#W 00 00 00 0C 21 0C 00 @

#W 00 00 00 04 4C 0C 00 @

#W 00 00 02 0D 4C 09 00 @

#W 00 00 0E 0F 2A 09 00 @

#W 00 00 06 01 34 09 00 @

Default window locations, sizes, and open directories and masks.

#P 03 04 000 C:\UTILS\FILE_VIE.WER\GUCK18\GUCK.TTP@ *.*@ @

This line installs Guck as the default file viewer. When the file is double clicked on, TOS checks all the #G, #P, #Y, and #F assignments below, and if the file doesn't match any of those, Guck is run and the file is loaded in. Notice that it is the first assignment in the INF file, and hence the last assignment TOS checks. If you moved this one down a few lines (say, underneath #F 03 04 000 *.TOS@ @ @), you'd never get any programs to run; everything (except the .INFs and archives) would be loaded directly into Guck!

#G 03 FF 000 *.ACC@ @ @

This line lets you run an ACC as a program, if the ACC is capable of it, by double-clicking on the ACC in any window.

```

#G 03 FF 000 *.APP@ @ @
#G 03 FF 000 *.PRG@ @ @
#Y 03 FF 000 *.GTP@ @ @
#P 03 FF 000 *.TTP@ @ @
#F 03 04 000 *.TOS@ @ @
#G 03 A0 200 C:\UTILS\ARC\STZIP26\STZIP.PRG@ *.ZIP@ @
#G 03 2F 200 C:\UTILS\ARC\TWOINONE.PRG@ *.ARC@ @
#G 03 2F 200 C:\UTILS\ARC\TWOINONE.PRG@ *.LZH@ @
#G 03 2F 200 C:\UTILS\ARC\TWOINONE.PRG@ *.ZOO@ @
#G 03 2F 200 C:\UTILS\ARC\TWOINONE.PRG@ *.ARJ@ @
#G 03 2F 200 C:\UTILS\LOADINF\LOADINF.PRG@ *.INF@ @

```

Executable programs and programs to run when files with the appropriate mask are opened. The second two digits of the third column of numbers ("00" of the "200") is the hex number of the function key assignment. "0A" would be F10.

Other parameters (for GTP or TTP) can be included before the last "@".

```
#D 0A 0A 000 @ *.*@ @
```

Default folder icon

```
#I 0B 0B 000 @ *.*@ @
```

Default file icon

```

#I 0C 0C 000 @ *.PR?@ @
#I 11 11 000 @ *.AC?@ @
#I 2E 2E 000 @ *.CPX@ @
#I 3A 3A 000 @ *.TTP@ @
#I 0D 0D 000 @ *.TOS@ @
#I 39 39 000 @ *.APP@ @
#I 39 39 000 @ *.PRG@ @
#I 12 12 000 @ *.ACC@ @
#I 52 52 000 @ READ*.*@ @
#I 52 52 000 @ *.ME@ @
#I 2C 2C 000 @ *.INF@ @
#I 13 13 000 @ *.RSC@ @
#I 3B 3B 000 @ DC*.*@ @
#I 1B 1B 000 @ *.HLP@ @
#I 6F 6F 000 @ G+*.*@ @
#I 3C 3C 000 @ DATADIET.*@ @
#I 46 46 011 @ WORDUP.PRG@ @
#I 49 49 004 @ QVIEW.PRG@ @
#I 51 51 000 @ SUPER*.PRG@ @
#I 63 63 002 @ PGS2.PRG@ @
#I 70 70 000 @ WARP9_ST.PRG@ @
#I 73 73 000 @ DFORMAT.PRG@ @
#I 3D 3D 00A @ SQUISHII.APP@ @
#I 6D 6D 001 @ TWOINONE.PRG@ @

```

File icons and function key assignments. Notice these act as filters also, and hence files are applied from the bottom up. For example, SQUISHII.APP is assigned icon #3D, but a file called SQUID.GHU doesn't match any filter, it would be assigned the default icon #0B.

```

#D 4E 4E 000 @ TELECOMM@ @
#D 3E 3E 000 @ AUTO@ @
#D 37 37 000 @ CLIPBRD@ @
#D 53 53 000 @ ACCS@ @
#D 55 55 000 @ CPX@ @

```

```
#D 35 35 000 @ TEXTE@ @
#D 36 36 000 @ BILDER.PAD@ @
#D 36 36 000 @ MYDRAW@ @
#D 36 36 000 @ META@ @
#D 36 36 000 @ GRAPHICS.Z@ @
#D 4F 4F 000 @ GAMES@ @
```

Folder icon assignments. Again, assignment is from bottom to top.

```
#M 00 00 00 FF A FLOPPY DISK@ @
#M 01 00 02 FF C BOOT@ @
#M 02 00 02 FF D PUBLISHING@ @
#M 03 00 02 FF E FILES@ @
#M 04 00 02 FF F PROGRAMMING@ @
#M 05 00 02 FF G GRAPHICS@ @
#M 06 00 02 FF H TELECOM@ @
#M 05 01 05 FF M RAM DISK@ @
```

Drive icon, letter, position, and label assignments.

```
#O 06 01 08 FF PRINTER@ @
```

Printer icon, position, and label assignment.

```
#T 07 00 09 FF GARBAGE@ @
```

Trash icon, position, and label assignment.

```
#X 07 01 7B FF C:\UTILS\RECV_CAN\RT.PRG@ TRASH CAN@
```

Executable file on the desktop. Icon, position, and label assignment.

```
#V 04 01 3E FF C:\AUTO\SUPERBT\*.INF@ DESKTOP@
```

Directory folder on the desktop. Position, icon, pathname, and label assignment.

Comments

Notice if a file is clicked on and is not a *.INF or one of the listed archival types, TOS then checks to see if it is either TOS, TTP, GTP, PRG, or APP (in that order), and if so executes it. If it is not one of these, the remaining installation for types *.* "catches" it, so GUCK will be run and the file will be loaded in. This then "replaces" the desktop [SHOW | PRINT |CANCEL]. Remember that each of these acts as a filter and that TOS applies them from *bottom* to top; so when you double-click on a file, TOS first checks to see if it is *.INF, and if so runs LOADINF.PRG, then it checks for type *.ARJ, then... through the executables (TOS, TTP, GTP, PRG, APP in that order), and finally if the file wasn't one of those, TOS type *.* (that's everything left) is loaded into GUCK.
