

## SIMION's GUI

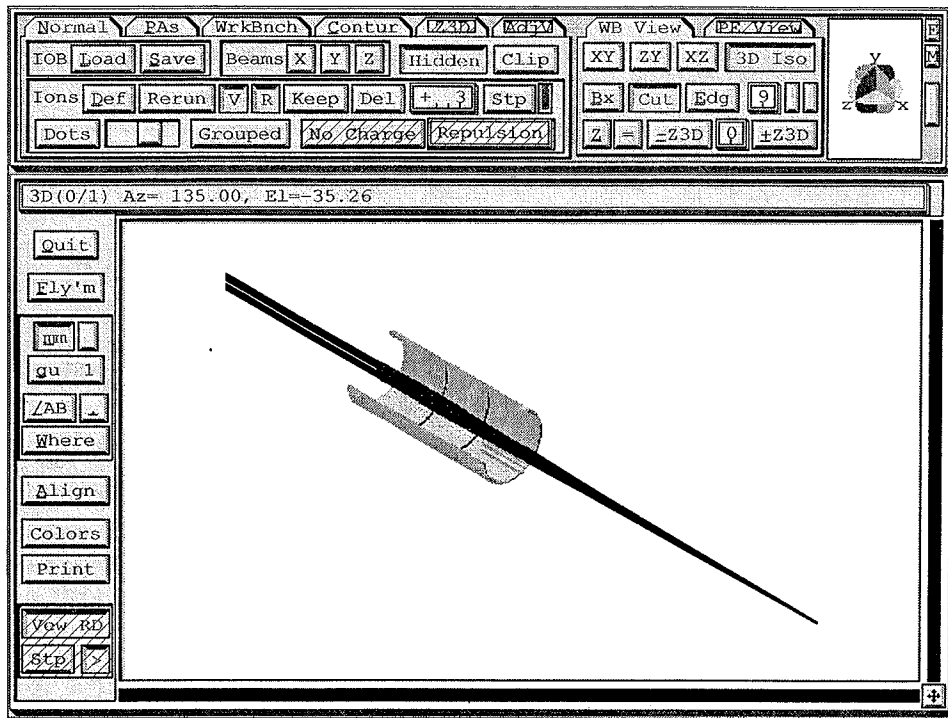


Figure F-1 Example of a GUI screen in SIMION

## Introduction

SIMION makes use of a *Graphics User Interface - GUI* originally created by the author for developing data systems (as well as *SIMION 6.0 – a DOS version*). This GUI has gradually evolved into a powerful platform for applications like SIMION 7.0.

Nevertheless, the decision to keep the GUI when porting SIMION into the Win32 environment (*Windows 9x, NT*) was a difficult one. In the final analysis the author felt that the GUI's performance (*efficient use of objects*), features (*e.g. high performance objects – panels and windows*), and continuity issues (*retaining consistency with SIMION 6.0*) made it a prudent choice.

SIMION 7.0 is actually a stealth-port of an enhanced SIMION 6.0 into the Win32 environment (*e.g. Windows 95/NT*). Windows thinks this is a true-blue Windows 9x/NT application while the GUI knows better. The bottom end of the GUI is fully Win32. However, the top end is SIMION's traditional GUI interface. You might want to think of this as a GUI application cloaked as a Win32 application.

This appendix is devoted to acquainting you with SIMION's GUI and its many useful features.

### An Overview of the GUI and Win32 Marriage

---

The following provides a brief overview of what in the GUI has changed and what has not for those familiar with SIMION 6.0.

#### Win32 Resources Utilized

---

The GUI makes use of the Windows GDI and device drivers. This means that SIMION appears in a window, uses the normal Windows video drivers and messaging system. The program also allows access to Windows printers, metafiles, and clipboards.

SIMION 7.0 makes full use of Win32's virtual memory (*2 Gigabytes*). However, there is no substitute for RAM if you want to run at faster than a glacial rate. SIMION 7.0 also makes full use of physical RAM (*SIMION 6.0 had a practical 64 MB physical RAM limit without heroics*).

The GUI's file manager has been modified to support Win32 file naming conventions including long filenames.

#### GUI Features Retained

---

The entire GUI user interface has been retained. Thus SIMION's look and feel is unchanged although it is a Win32 application. The GUI printer drivers have been retained and are available in addition to full support for Windows printers, metafiles, and clipboards. The reason for this is that the GUI printer drivers are very general and useful when the highest quality output is required (*they use floating point graphics in contrast to Windows integer graphics output*).

#### Use Differences

---

The following is an important list of program differences:

##### Viewing and Sizing SIMION's Window

SIMION always comes up as a **full screen adjustable** window (*not maximized*). You can resize the window if you like by normal Windows *methods* (*by dragging the edges or clicking the maximize button*). SIMION will automatically adjust its objects to fit the currently defined window size. Scroll bars will appear if you try to resize the window below its minimum client window limits (*640 in x by 480 in y*).

**Note:** SIMION will not resize its objects (when its window is resized) if a Print Options screen is active to prevent havoc with annotation objects.

##### Accessing SIMION Via Mouse and Keyboard

Access to SIMION by mouse and keyboard **requires** that its window have input focus (*it is the foreground window*) **and** that the cursor is within the client area of the window (*a GUI cursor is visible*). Move the mouse cursor into SIMION's client area (*GUI screen area*). If the cursor doesn't automatically change from a Window's cursor to a GUI cursor, click the left mouse button to establish input focus. Now the cursor will be the GUI cursor.

Notice that many buttons have their first character underlined. This means you may access them directly from the keyboard (*when the cursor is in the window's client area*). When you move the cursor out of the client area the GUI cursor will be changed to the Window's cursor

and the button underlines will be removed to tell you that the buttons are no longer keyboard accessible.

***Remember:*** Keyboard access requires that SIMION's window has both input focus and that the cursor is in the client area (GUI cursor visible).

### Long Filename Issues

All SIMION files (e.g. .JOB) now support long filenames as well as the MSDOS 8.3 format. This has required changes in the format of certain display screens to allow display of longer filenames. The standard filename display width is 27 characters. SIMION will show names longer than 27 characters as truncated (e.g. This is a ... very long file.pa). **If you want to see the full name and/or short name (MSDOS 8.3 format) of the file, point the cursor to the display panel or button and click the FI key.**

SIMION requires that file systems must support both the long filenames and MSDOS 8.3 short names (e.g. dual filenames). This means that SIMION 7.0 will only work properly with FAT, protected mode FAT, and New Technology file system (NTFS). **SIMION 7.0 will not work properly with the High-performance file system (HPFS). This means that SIMION should work on all Windows 95 and those Windows NT installations that do not employ HPFS. If you are connected to a network, SIMION will not work properly on any network drive that uses HPFS.**

Whenever you edit a file, via one of the many edit buttons within SIMION 7.0, the editor (EDY by default) is normally passed the file's short name (8.3). This allows MSDOS editors like EDY to support SIMION 7.0. When you replace the file, *without changing its name*, the long filename will be retained. If you want to create a long filename file with EDY, create the file and save it with a short name (8.3). Now, within the GUI File Manager, depress the file's button, click the **Other** button, rename it using the **Rename Marked file** ioline object, and press the **Enter** key. *See Appendix B or H to see how to pass long names to editors.*

If you want to switch to .PAs with long filenames within an existing .JOB, you must first make long filename copies of the affected .PAs and .PRG files. The following MSDOS command (accessed via the GUI File Manager's **Other** button) demonstrates how:

```
COPY SHORT.P* "Some Long Name.p*"
```

Note: The quotation marks (" ") are required only if the long filename contains spaces.

Now that you have made long filename copies of the desired .PAs, load the existing .JOB file into SIMION, click the **PAs** tab, select each .PA to replace, and use the **Rpl** button to replace it with its long filename counterpart. Be sure to save the .JOB file after all the changes have been made. Now use the GUI File Manager to delete the replaced short name (8.3) files.

### Exiting from SIMION

You can exit from SIMION in the Normal GUI manner (*Esc* or *Exit* button) or by using any Windows close option (e.g. the *close* button).

### Multitasking with SIMION

You can run SIMION while other processes are also running. This means you can start a refine, minimize SIMION, and continue with some other project. SIMION will continue to refine, fly ions, or what have you in the background. While more than one copy of SIMION can be running at the same time it is usually not recommended because the adverse impact that this can have on computer performance (unless you have a multiple processor system).

## The Philosophy of SIMION's GUI

---

SIMION's GUI differs somewhat from the Windows GUI. However, some of its more advanced features point in the general direction GUIs appear to be evolving. The philosophy behind this GUI can be summarized as follows: *Make it familiar, easy to learn, easy to use, and powerful.* The illustrations below serve to demonstrate this philosophy:

### First: Make it Familiar

---

The GUI makes extensive use of familiar objects like buttons, sliders, and the like (Figure F-2). These objects are combined to resemble the familiar interfaces of the electronics devices used around us.

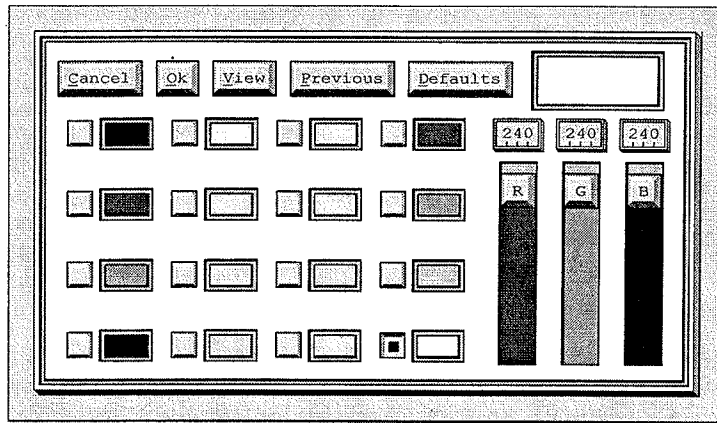


Figure F-2 The GUI's Palette Screen

### Second: Make it Easy to Learn

---

Since people don't typically read manuals, each object on the screen has its own help screen to explain its function and use. Just point the cursor at an object and press the <F1> help button (Figure F-3). Most objects also provide additional levels of help via the More and Object Help buttons on their help screen. Figures F-4 and F-5 below illustrate using these additional buttons to gain access to these additional levels of help:

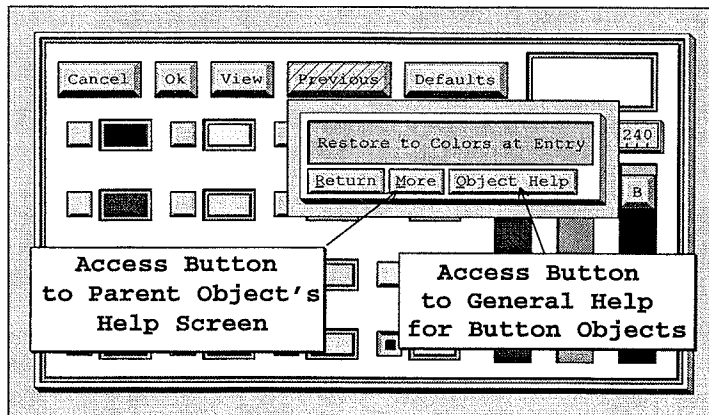


Figure F-3 The GUI's Help Screen facility

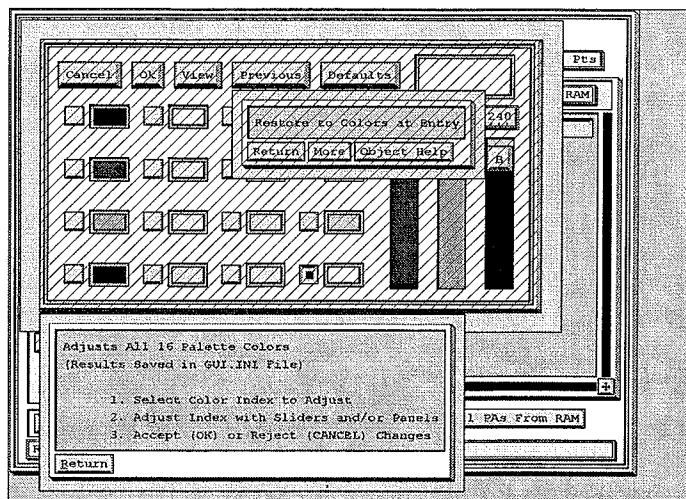


Figure F-4 Example of the use of the More button

### The More Button

The **More** button (when provided – Figure F-4) gives direct access to the help screen of the object's parent (and so on up the parental hierarchy). SIMION's *help strategy* is designed to go from the *specific to the general* (a more natural direction).

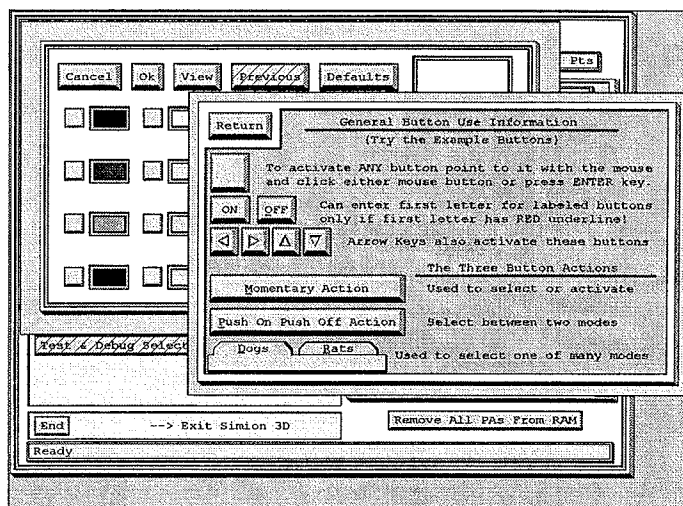


Figure F-5 Example of an Object Help Screen

### The Object Help Button

The **Object Help** button (when provided – Figure F-5) accesses a *help screen for the selected object class* (e.g. button class). The object class help screen explains the features of the class of objects and how to use them. *Usable example objects of the class are provided on the help screen to serve as learning aids (victims).*

## Third: Make it Easy to Use

---

Many features like: *Esc key aborting*, *assisted mouse pointing*, and *direct keyboard access* capabilities assist in the GUI's ease of use:

### The Esc Key - Your Yellow Brick Road

---

If you're lost or just want to stop something in progress (e.g. a printout), hit the <Esc> key. *Each time you press the <Esc> key the GUI will stop what it's currently doing and back out a layer.* Of course, there is built in protection to prevent you from aborting right out of something important (e.g. like the program itself).

### Assisted Mouse Pointing

---

*The GUI knows where the objects are on your screen and assists you in pointing to them.* Visual and audio cues (*object blinking*, *cursor changes*, and *sounds*) are provided to further reinforce pointing feedback.

In query screens, the *GUI moves the cursor to the default object (or answer button)*, and then *returns the cursor to its starting point* when the query screen is removed.

If your inputs are currently limited to some special screen, *the GUI will either automatically limit (lock) your cursor's movements* to that area *or block recognition (e.g. no blink, click, or cursor change)* of the inaccessible objects. This prevents you from *falsely* assuming that you might have access to objects outside the active area.

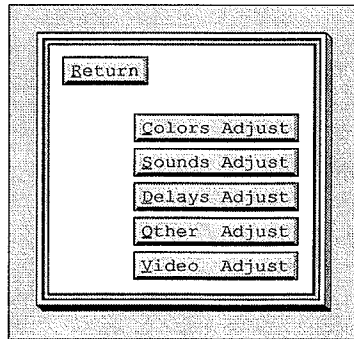


Figure F-6 Direct Keyboard Access to Buttons

*The result is easier and faster program access via the mouse.*

### Direct Keyboard Access to Buttons

---

While mice may be nice, *interfaces that also support simple keystroke entry provide the fastest program access for the experienced user.*

In non-GUI DOS versions of SIMION a potential array was saved by pressing the <S> key (for Save command) entering the name of the file and then pressing <Enter> to save it.

**The same keystrokes can be used in SIMION 7.0 (Figure F-6).** This is because many (though not all) labeled buttons can be directly accessed by entering the first letter of their label on the keyboard (e.g. <S> for Save).

**Whenever a labeled button is currently keyboard accessible the GUI will automatically place a red underline below the first letter of the button's label.**

*Keyboard accessibility of labeled buttons depends on their order of creation and on the location of the mouse cursor. The most recently created buttons (on highest level screen) have the highest priority for direct keyboard access. However, if the cursor is in an ioline object (e.g. used for entering a filename), all keyboard entries are processed by the ioline object.*

*Whenever a labeled button is not currently keyboard accessible the GUI will automatically remove the red underline from its first letter (e.g. when the cursor enters an ioline object).*

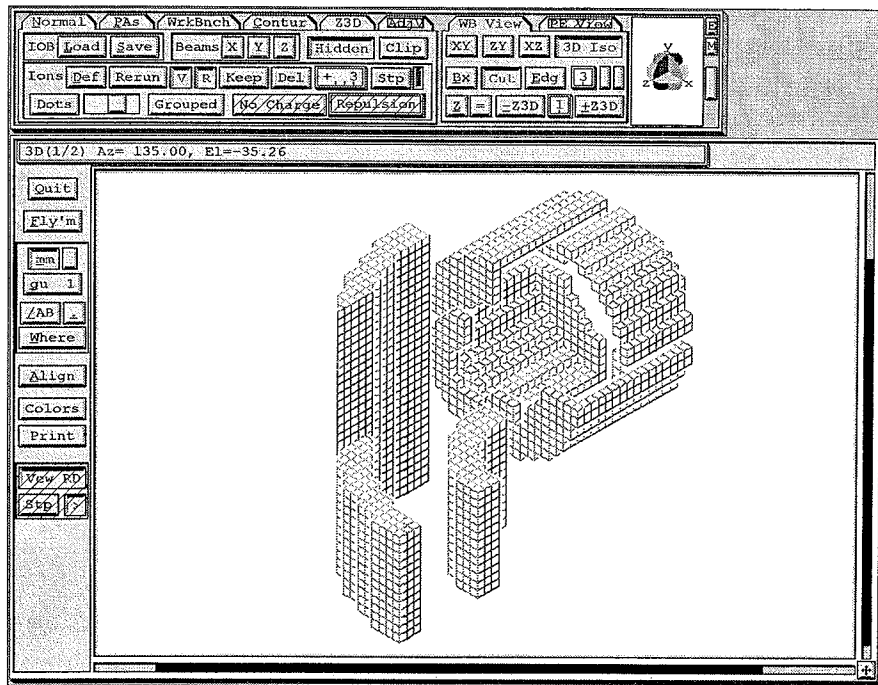


Figure F-7 Example of one button GUI windows

#### Fourth: Make it Powerful

The GUI includes innovative features like *one button windows*, *trapped cursor scrolling*, and *bi-directional memory zooms* that really help make SIMION 7.0 easy to use (Figure F-7).

##### One Button Windows

While other GUIs control their windows with lots of little objects *controlling horizontal and vertical motions separately*, this GUI uses just one button for *full bi-directional control*.

Drag this button with the **left** mouse button depressed and you have *full 2D scrolling capability*. Hold down the <Ctrl> key while you drag the mouse and the *view dynamically zooms*. Hold down the <Alt> key while you drag the mouse for *true 3D pointing and volume cutting* capabilities in SIMION.

Drag this button with the **right** mouse button depressed and you have all the functions above in *full page (world) view*.

Trapped Cursor Scrolling

If you hold down the <Ctrl> key when the cursor is within the view area of a window the window will automatically scroll the view when the cursor hits a view boundary (*2D demand scrolling*).

Quick Bi-directional Memory Zooms

To zoom in, just *mark the desired zoom area* by dragging in the view area with the **left** mouse button depressed and then click the **right** mouse button to zoom. If an *area is not marked*, clicking the **right mouse button** in the view area will *zoom out* to the prior zoom level (*up to 10 levels remembered*). However, if either <Shift> key is *depressed* clicking the **right** mouse button with no area marked will *zoom in* to the next zoom level (*if defined*).

Standard GUI Objects

The GUI screen is composed of layers of objects. Some of these objects are simple surfaces, others are standard functional objects like buttons, while others are custom objects designed for special tasks within an application (*e.g. the view orientation control sphere in SIMION's view function*). The discussion here will be limited to standard functional objects: **Buttons, Sliders, Iolines, Panels, Selectors, and Windows**. Each object's general help screen will serve as the basis for explaining it:

Button Objects

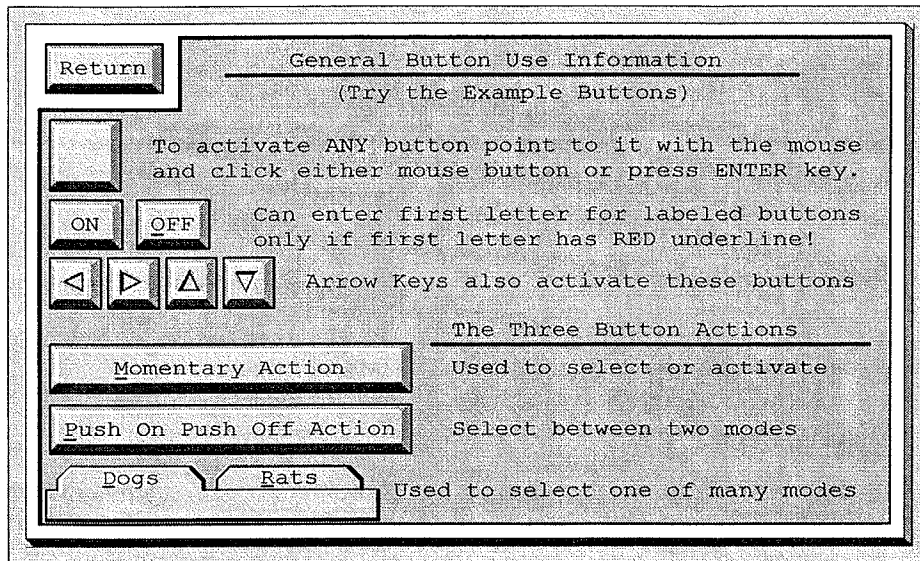


Figure F-8 Object help on button objects



## Slider Objects

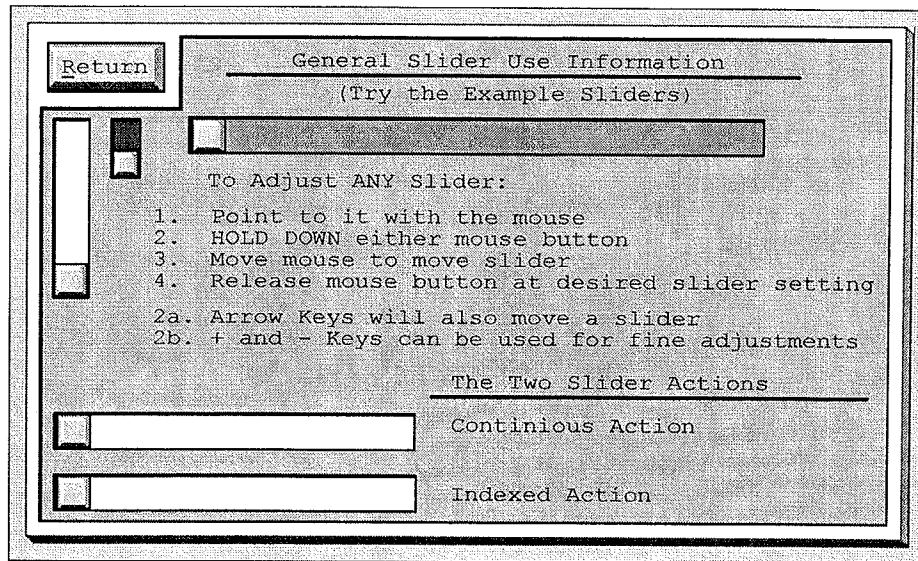


Figure F-9 Object help on slider objects

## Ioline Objects

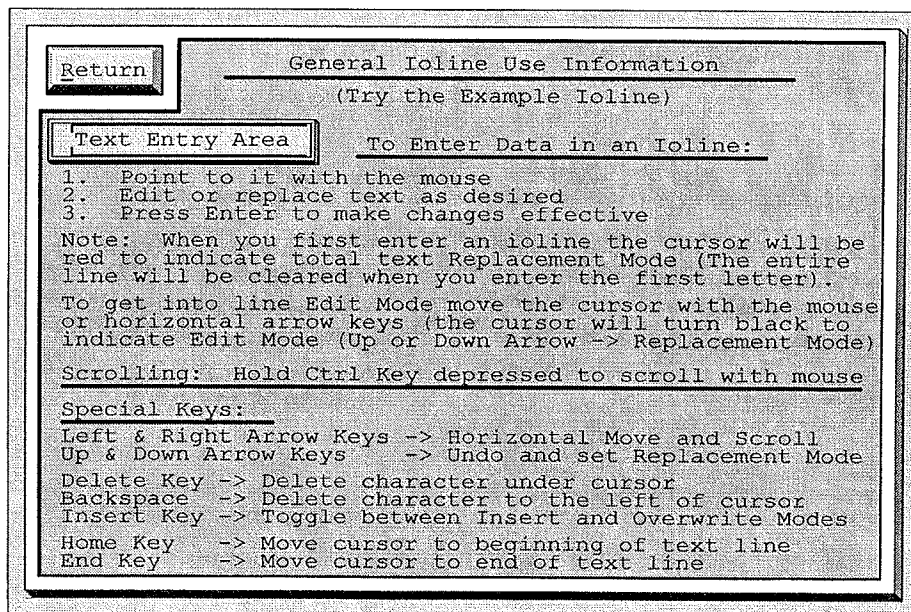


Figure F-10 Object help on ioline objects

Panel Objects

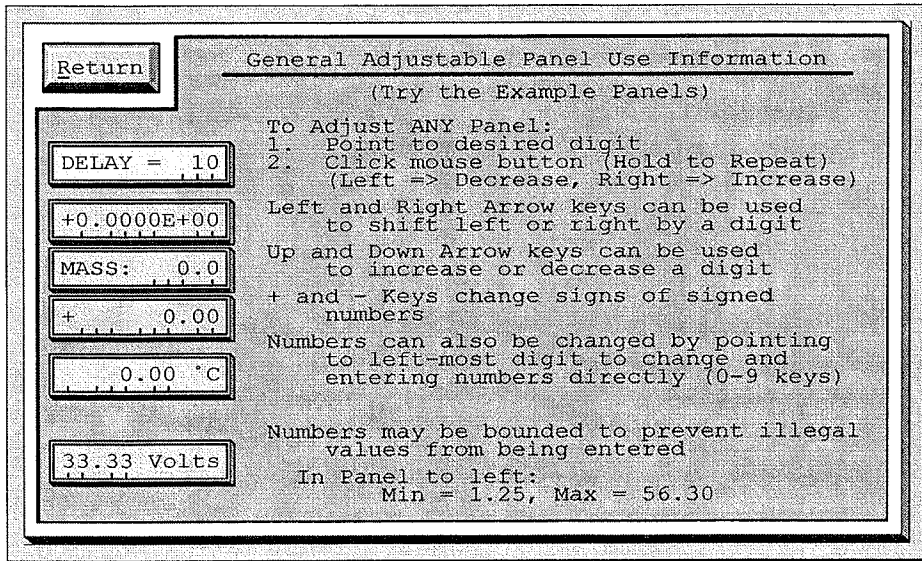


Figure F-11 Object help on panel objects

The Two types of Panel Objects

There are really two types of panel objects: *Adjustable and non-adjustable*. The *adjustable* variety have *small vertical lines below each digit* that can be adjusted (as in the example adjustable panels above – Figure F-11). Non-adjustable panels are used to display strings of text (e.g. status and warning indicators).

Selector Objects

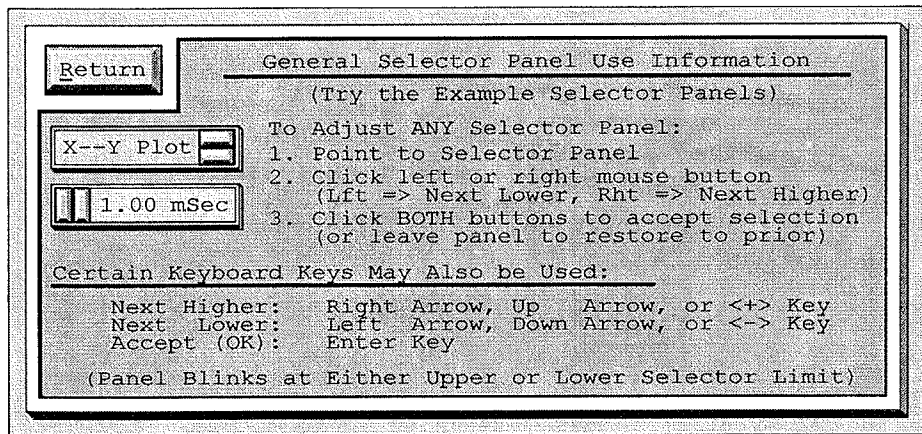


Figure F-12 Object help on selector objects

Note: Some selector objects automatically accept the selection (e.g. Print mode selector). Automatic acceptance occurs to speed up the process and when no adverse consequences would occur from user fiddling with the selector.

## Window Objects

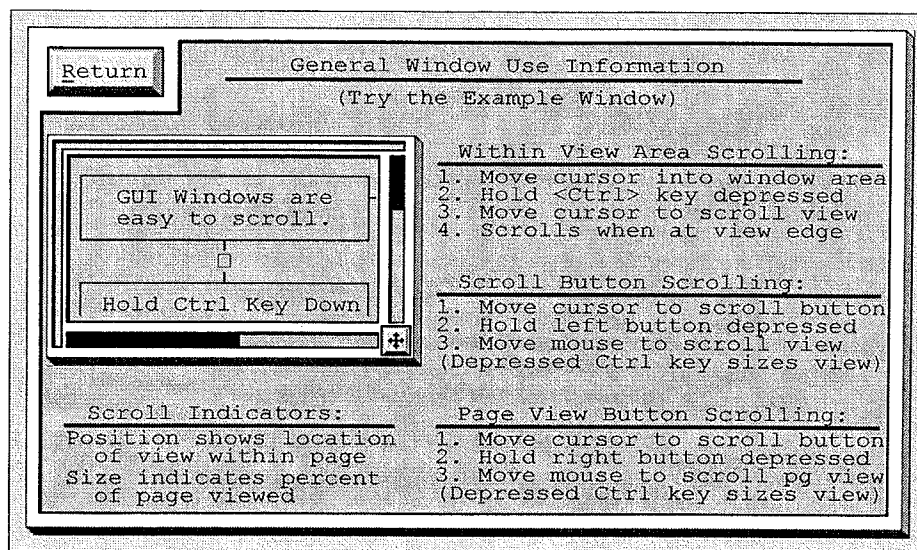


Figure F-13 Object help on window objects

*Note:* The GUI uses line drawing and other special characters in the IBM extended character set to draw text boxes and arrows. Unfortunately, the **Linedraw.ttf** font used by the GUI for Windows printing doesn't support some these characters (e.g. arrows) in its extended character set. *Thus you see a box outline for an arrow character in the illustration above (Figure F-13).*

## Normal Verses Page View

The GUI window button in both text and graphics windows provides access to two views: *Normal view* and *page view*. *Scrolling, Zooming, and 3D pointing* function in either view.

### Normal View - (Current Position and Magnification)

Is accessed by activating the window button with the **Left** mouse button.

### Page View - (World View)

Is accessed by activating the window button with the **Right** mouse button. The area of the normal view is shaded or outlined. *Page view is useful for fast scrolls and zooms.*

## Text Windows (text only windows)

### Scrolling Text Windows

**Scrolling is supported** if the page area is larger than the visible area.

**Window Button Scrolling:** Depress (and hold down) the window button and move the mouse in the direction you want the view to scroll.

**Cursor-In-View Demand Scrolling:** Hold <Ctrl> key depressed when cursor is in window's view area. View will scroll when cursor hits a view boundary.

**Graphics Windows (lines and other graphic elements)**

---

**Scrolling Graphics Windows**

---

**Scrolling** is supported as in text windows (*above*).

**Zooming Graphics Windows**

---

**Window Button Zooming:** Hold <Ctrl> key depressed and depress (*and hold down*) the window button. Move the mouse *away to zoom out, inwards to zoom in*.

**Cursor-In-View Zoom in:** Mark area to zoom into using **left** mouse button. Click **right** mouse button to zoom in (*ten level memory zooming supported*).

**Cursor-In-View Zoom out:** Don't mark area. Just click **right** mouse button to zoom out.

**Cursor-In-View Zoom back in:** Don't mark area. Hold either <Shift> key depressed and click **right** mouse button to zoom in (*if defined*).

**3D Pointing (Isometric 3D Views only)**

---

**3D Isometric Cut Plane Control:** Hold the <Alt> key depressed and depress (*and hold down*) the window button. Move the mouse in the *appropriate isometric direction* to select an isometric cut plane and move it in the desired direction.

**3D Cursor-In-View Pointing:** Hold the <Alt> key depressed with cursor in window's view area. Move the mouse (*with left button depressed*) in the *appropriate isometric direction* to move cursor in 3D and define the inner volume.

**Quick Draw Scrolling, Zooming, and 3D Pointing**

---

You can *temporarily* tell the GUI to eliminate (*or at least minimize*) redrawing. To activate *quick draw* just depress the *other* mouse button during the *scrolling, zooming* or *3D pointing* process.

**GUI Customizing Screens**

---

The GUI has five customizing screens that allow you to change *colors, sounds, delays, other items*, and *video resolution*. In SIMION, you gain access to these screens via the **Adjust** button on the main screen. *Any changes you make are retained (between sessions) in the C:\FILES.GUI\WGUI.INI*

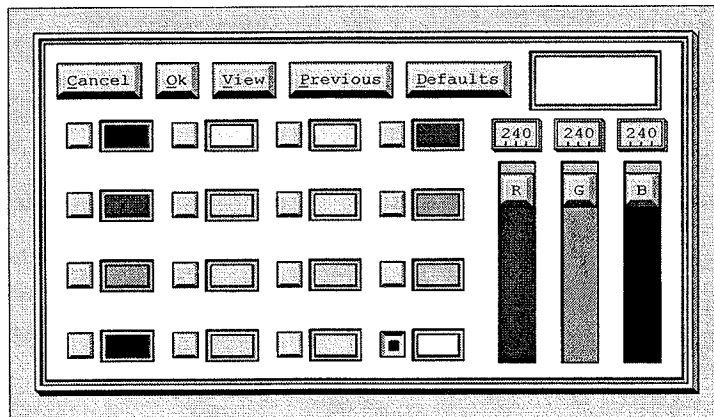


Figure F-14 Color Customizing Screen

and C:\FILES.GUI\WVIDEO.INI (*Video options only*) files.

### Color Customizing Screen

---

This screen sets the 16 primary colors used by the GUI (*Figure F-14 above*). The SIMION **Modify** and **View** functions as well as the Print Options Screen provide a *Color* button to allow you to change colors for viewing and printing preferences (*e.g. for emphasis*).

The **View** button has been added to the Win32 GUI to force a client area redraw so that you can see the impact of the color changes you are considering.

*Note:* The three shades of brown (*rightmost column*) are used for simulating surface illumination

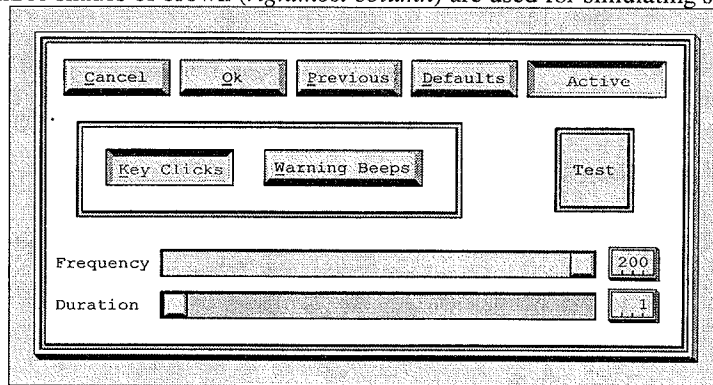


Figure F-15 Sound Customizing Screen

in 3D isometric views.

### Sound Customizing Screen

---

The *sound customizing* screen is used to adjust the *warning beeps* and *key clicks* (*Figure F-15*). You can adjust each for *frequency* and *duration*. Moreover, there is an **Active** button (*upper right*

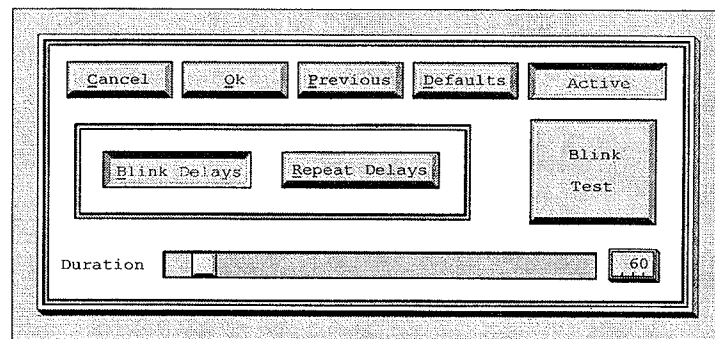


Figure F-16 Delay Customizing Screen

*corner*) that can be used to turn off clicks and/or beeps.

### Delay Customizing Screen

---

The delay screen is used to control the *blink* and *repeat delays* (*Figure F-16*). When the cursor enters an object that is capable of action it normally blinks the object. The repeat delay determines

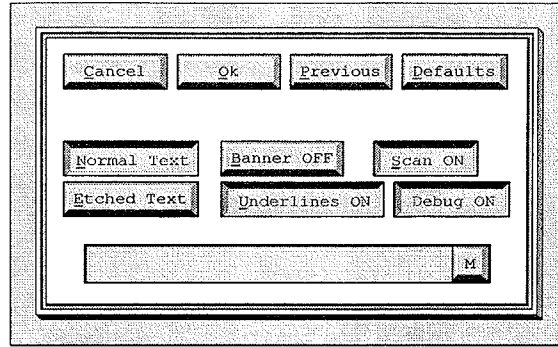


Figure F-17 Other Items Customizing Screen

how fast a momentary button object releases when activated by a keyboard entry. *Repeat delay also controls the mouse button repeat rate when used to adjust panel objects.*

### Other Items Customizing Screen

Two buttons (*on the left*) select between **normal** or **etched** text on button objects (*Figure F-17 above demonstrates normal text*). The **Underlines** button controls whether the GUI is allowed to tell you (*with red underlines*) if a labeled button is currently keyboard accessible (*buttons still remain keyboard accessible*). The **Scan ON/Scan OFF** button tells the GUI File Manager whether to *automatically scan* a drive for directories the *first time it is selected* in the current program session. The **Debug On/Debug OFF** button activates detailed GUI/SIMION debugging and internal error trapping

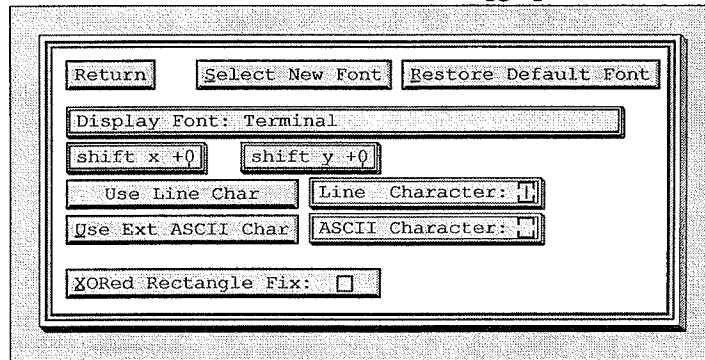


Figure F-18 Video Options Screen

(normally depressed – ON) The slider object adjusts mouse sensitivity: left (*low*) to right (*high*).

### Video Options Screen

This screen has been totally changed from SIMION 6.0 to support Windows video drivers and fonts. *SIMION will preserve video option changes between sessions.*

#### Selection of Screen Font

You have the option of selecting the Windows screen font to be used with SIMION. The default font SIMION tries to use is **Terminal** font with the **OEM/DOS** character set. If that is not available or you want to use another font you can click the **Select New Font** button and make your selection from the various fixed-pitch fonts available. SIMION requires that the

resulting font is 8 pixels wide. If the font and size you select is not available in this size SIMION will refuse to use it.

### Label Positioning Panels

Two label positioning panel objects are provided. These allow you to shift the location of the labels created with the currently selected screen font as may be required to locate them properly on buttons, panels, and etc.

### Using a Line Drawing Character

Most of the fonts other than **Terminal** and **MS LineDraw** do not have line drawing characters in their character set. This causes strange characters to appear in the GUI File Manager's directory tree pictures. You can substitute a fixed line drawing character. If you depress the **Use Line Char** button the currently defined fixed line drawing character will be used. The default is the | or vertical line. However you can use the ioline object to the right to specify an alternate character if you desire.

### Replacing Other OEM characters

SIMION occasionally makes use of certain OEM characters like the degree character “°” if the selected font's character set doesn't have an equivalent character a strange symbol may appear here and there. If you depress the Use **Ext ASCII Char** button a space (*by default*) will be substituted for all extended ASCII characters (*except line drawing characters*). You can use the ioline object to the right of the button to select an alternate substitution character.

### The XORed Rectangle Fix Button

The **XORed Rectangle Fix** button (*on the bottom of the screen*) is used to compensate for a common Windows video driver defect. Many Windows video drivers do not draw xored rectangles properly. This makes your GUI target cursor grow hair and ion dots look like triangles or worse. If the square on the button looks weird depress the button. Assuming the square on the button now looks OK, your video driver has the problem and the GUI is now compensating for it.

Note: Some video drivers appear to draw the cursor faster with the **XORed Rectangle Fix** button is depressed. If your GUI cursor movements are a bit jumpy depress this button to see if cursor drawing improves.

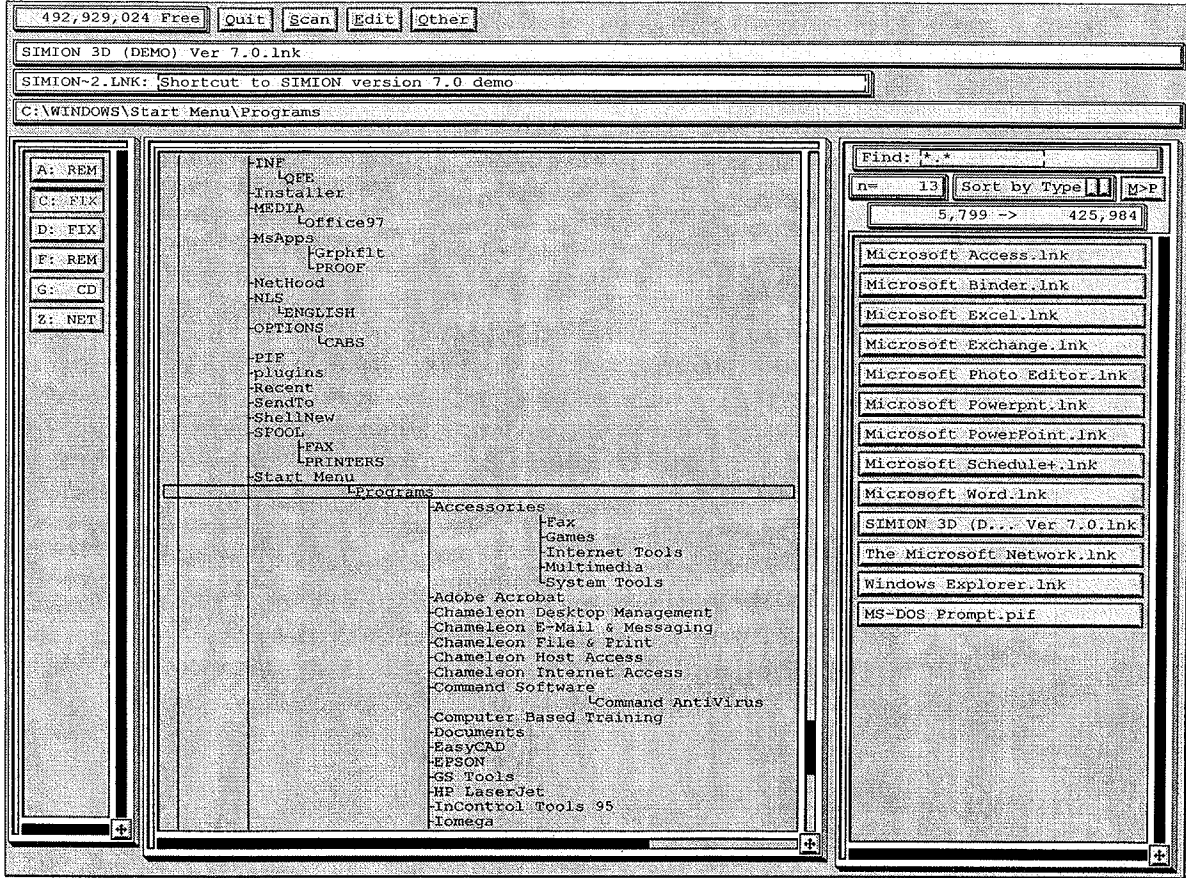


Figure F- 19 GUI File Manager's screen

## The GUI File Manager

### Introduction

The GUI's File Manager (*Win32 version*) is illustrated above (*Figure F-19*). This GUI feature is used extensively within SIMION for loading and saving files. However, you will find that it is a *powerful tool* for doing all sorts of things. The information below only gets you started. *Use the <F1> help key on the GUI File Manager's objects to learn the details.*

### A Quick Tour of the File Manager Windows

There are three windows in the screen above:

#### Drive Display Window

The *leftmost window* is used to view and select the *current drive* (e.g. C:). Note: The type of drive (e.g. FIX for fixed disk) is also displayed. Note: A drive is normally scanned for directories the *first time* it is *selected* in the *current program session*. Use the **Scan ON/Scan OFF** button (*Other Options Customizing Screen*) to suppress auto-scanning.

*A drive is selected by clicking its button.*



## Directory Tree Window

The **middle window** shows the **complete directory tree** for this drive. The **current directory is marked** (e.g. C:\WINDOWS\Start Menu\Programs) and its path displayed above the window. *Note: Long name paths are now displayed and supported.*

*A directory is selected by clicking on it.*

When you move the cursor to a directory its size and its tree size will automatically be displayed on the cursor. The cursor numbers 25.3M(63.0K) mean that the directory's tree (*the directory and its sub-directories*) requires 25.3 megabytes, and the directory itself only requires 63 kilobytes of disk space. *Click the Scan button first if files have been deleted or added since the last directory scan to insure the directory sizes are accurate.*

*Note: The page view capability of the GUI is helpful leaping about in large drives.* Move the mouse button to the directory tree's window scroll button (*lower right corner*). Hold down the **right** mouse button. A view of the entire drive's directory tree will appear as a line image in the center and its first level of directories will appear on the left edge. Move the mouse (*right mouse button still depressed*) up and down to move quickly through the drive's directory tree. When a directory's name is shown in **red** it will be visible when the mouse button is released. If you want to view more than one directory level, hold down the **Ctrl** (*zoom*) key too. Move the mouse *down* or to the **right** (*slowly*) to *increase* the directory tree levels displayed. Move in the other directions to *decrease* the levels displayed. *Page view is a quick way to find and jump to the desired directory in a very large drive.*

## Files Found Window

The **rightmost window** contains a listing of the **files found** in the current directory. This list displays the file's long name truncated if necessary to fit in a 27 character window (e.g. *SIMION 3D (D... Ver 7.0.lnk)*). If you point the cursor to a file's button its complete long name will automatically appear in the top panel object and its short name will appear at the beginning of the memo ioline object (*the next object down*). The GUI's page view capability described above for directory tree windows can also be used to quickly locate files dragging the files found window's scroll button with the mouse **right** button depressed.

*One or more files are selected and de-selected by clicking on the files' buttons.*

There are five objects at the top of the files found window. The **topmost** is an *ioline* that contains the **current search criteria** (e.g. *\*.\* - you can change it*). In this case the file search was for all files (*\*.\**). SIMION often changes this parameter to find selected files (e.g. *.PA?*).

The next object is a display only panel that shows the number of files found that match the search criteria (e.g. *n = 13 in this example*).

To the right is a **selector object** that determines the current **file sorting order**: By type (*by extension then name*), by name (*by name then extension*), by creation date (*youngest on top*), or by file size (*largest on top*).

Last on the line a small **M>P** button. This button is used to **output file memos** for the **currently found files** to a printer/file (*ASCII image only*).

Just below is a display **panel** that shows two numbers the **actual total size required** *verses* the **disk space used by the found files**. *Note: The number on the right includes the wasted cluster space (5,799 bytes requires 425,984 bytes of disk – large disk allocation clusters (FAT 16) – not efficient for small files).*

*Note: If you move the cursor to a file's button, the current directory's path display (panel object just above windows) will switch to displaying the directory information for that file (e.g. size, date, and attributes: A = archive, R = read only, H = hidden, S = system).*

### File Memos

---

The GUI File Manager provides a powerful file memo capability. Each file can have a memo (*up to 70 characters long*) associated with it (*memos are saved in the MEMOINFO.GUI file in the current directory*). **Just above the current directory's path display object is a long ioline for creating and viewing file memos.** In the example above, the file memo for the file **SIMION~2.LNK** (*File's short name appears first in line – long name is above*) is displayed.

#### Viewing File Memos

---

To view a file's memo (*if any*) just **point to its file button** in the *found files* window.

#### Creating File Memos

---

To create (*or edit*) a memo for a file, **point the cursor to the file's button**. Now **move** the cursor to the **left off the file's button**. If you are successful, the **file's short name should still be in the memo ioline**. Now use the ioline object to **enter or edit the memo**.

#### The GUI File Manager and Memos

---

The GUI File Manager automatically manages your memos for you. If you use the file manager to copy a file with a memo its memo will be copied too. Likewise, deleting a file with the GUI File Manager will delete its memo too. **Moral: If you use memos use the GUI File Manager.**

Memos for files are saved in a **MEMOINFO.GUI** memo file in the current directory.

### The Top Row of Objects

---

#### The Disk Free Space Ioline Object

---

The top left panel object displays the **free space in bytes** of the **currently selected drive**.

#### The Cancel (or Esc) and OK (or Return) Buttons

---

The **Cancel (or Esc)** button is used to abort the file manager. You are **returned to the drive and directory at the time you entered the file manager**. The **OK (or Return)** button returns the filename you selected (*if requested*) and **stays in the currently selected drive and directory**.

#### The Scan Button

---

The **Scan** button is used to **rescan the current drive for directories and directory sizes**. If you create or delete directories **outside** of the file manager (*e.g. in MSDOS or a normal Window application*) or change disks in a drive, these changes **may not** be reflected in the directory tree displayed. Moreover, if files have been added or deleted, the **Scan** button should be clicked to insure that the directory sizes displayed on the cursor (*described above*) are accurate. **Scan** is **more** important when **Scan OFF** is active (*see Other Items Customizing Screen above*).

#### The Edit Button

---

The file manager provides an **Edit** button when the use of a file editor is allowed. To edit a file, select its button (*leaving it depressed*) and then click the **Edit** button. **The GUI uses the EDY editor by default**. If you prefer to use another editor, see Appendix H on EDY to see how to connect SIMION to an alternate editor.

### The Other Button

The **Other** button supports a wide variety of operations from file and directory copying to directory renaming. *Note: For operations on files, select them (depress their buttons) before clicking Other.*

### The File Ioline Object

The **File** ioline object (*when provided*) is used for entry of a file name. Note: You can point the cursor to the file's button and click *a* mouse button (*click both for select and OK exit*).

## Some Examples of File Manager Operations

### Deleting One or More Files

Select File(s) by depressing their buttons, click **Other** button, and click **Delete Marked Files** button (*you will be asked if you're sure*).

### Copying or Moving One or More Files

Select File(s) by depressing their buttons, click **Other** button, click **Copy or Move Marked Files** button, and select copy *or* move. Now select the destination drive and directory and click the **Proceed** or **Cancel** button. Source file attributes (*e.g. ready only*) are copied to the destination files (*see changing file attributes below*). *Note: The file manager will always ask permission before overwriting any files.*

### Renaming a File

Select File by depressing its button, click **Other** button, edit file's name in the **Rename Marked File** ioline, *and* hit <Enter>.

### Changing Marked File(s) Attributes

Select File(s) by depressing their buttons, click **Other** button, and click the **Change Attributes of Marked File(s)** button. Use the buttons or sliders to obtain the desired changes in file attributes. This option is a fast way to remove read-only file protection when needed.

### Changing All File(s) Attributes in Directory or its Tree

Click on the directory or base directory of the selected directory tree (*Don't Select any File(s)*), click **Other** button, and click the **Change ALL File Attributes in Directory or Tree** button. Select **All Files in Current Directory's Tree** (*default*) or depress button to select **All Files in Current Directory (ONLY)** Use the sliders to obtain the desired changes in file attributes. This option is a fast way to remove read-only file protection when needed from files or directories of files you may have copied from a CD ROM. Note: File attributes (*e.g. the CD-ROM read-only attribute*) are retained during file copies (*discussed above*) and directory copies (*discussed below*).

### Inserting a New Directory

Click on the new directory's parent directory. Now click the **Other** button, and enter the new directory's name in the **Insert Directory** ioline and hit <Enter>.

### Deleting an Existing Directory

Click on the directory, click the **Other** button, and click the **Delete Current Directory** button. If the directory is not empty you will be warned. *If you persist (be very careful) the file manager can be convinced to delete the directory, its files, and all of its sub-directories.*

### Renaming a Directory

Click on the directory, click the **Other** button, and edit the directory's name in the **Rename Directory** ioline and hit <Enter>.

### Copying or Moving One or More Directories

You can also copy or move one or a group of directories. Click on the root of the directory tree to copy/move. Click **Other** button and then click the **Copy or Move Directory** button. Select whether you want to copy *or* move (*delete after copy*) the directories. Now click on the directory that is to be the parent of the copied or moved directory tree, and click the **Proceed** button. You will be asked permission to overwrite any existing files.

You also have the option of copying a single directory. If the directory is at the end of a directory chain (*innermost directory*), you can copy or move it as described above. However, if the directory is somewhere else in the directory chain (*top to middle*) an **ONLY Copy Single Directory** button will appear on the Copy or Move Directories Option screen. If you depress this button, a single directory copy will be optioned. Note: The **Move Directories** button will then be blocked to prevent moving a single imbedded directory.

Entire disks can be copied by clicking on the drive's root directory (*e.g. D:\*). **Move directories** will be blocked, and the **ONLY Copy Single Directory** option is available. Note: The source root directory will be copied *into* (*rather than below*) the target directory. You will be required to confirm before an entire disk is copied.

### Accessing the MSDOS Command Prompt

At the bottom of the Other Screen (*click the Other button*), is the **Access MSDOS Prompt: EXIT to Return** button. If you click this button you will access a screen with the MSDOS prompt. You and execute programs and MSDOS commands from the prompt. When you are done type the word **EXIT** at the command prompt and hit <Enter> to return.