Printing Options

Introduction

SIMION provides powerful printing capabilities. ***These capabilities include:*** Output to Windows Printers, clipboard, metafiles, enhanced metafiles plus printer language drivers, color and B/W, image composition, and image annotation. The material below is provided to assist you in using your printer effectively with SIMION.

Note: If an Annotate and Print Screen is active, SIMION will not resize its objects if its window is resized to prevent havoc with annotation objects. Moral: Do any window resizing before to clicking the Print button.

Supported Printer Output Destinations

Windows Printer, Metafile, and Clipboard Support

SIMION 7.0 supports Windows printer, clipboard, metafile, and enhanced metafile output. This has been done by adding these as driver options to the Print Options Menu. These devices can be accessed for both pixel and vector printing (*discussed below*).

GUI Floating Point Printer Language Drivers

Note: The original GUI printer language drivers (*from SIMION 6.0*) have also been retained and are available ***in addition to*** support for Windows printers, metafiles, and clipboards. The reason for retaining the GUI printer drivers is that they are very general and useful when the highest quality output is required (*they use floating point graphics in contrast to Windows integer graphics output*).

The GUI printer drivers support the following printer languages: **PostScript (*B/W and Color*), PCL 5 (*B/W and Color*), HP-GL/2, and HP-GL.** ***These drivers support printer languages in lieu of specific printers***. If your printer supports one or more of these languages you have the option of using a GUI printer language driver in lieu of a Windows printer driver. These printer drivers often provide higher quality output and quicker printing than the corresponding Windows printer driver.

A Word About MSDOS Printer Connections

The GUI printer language drivers output through MSDOS channels, and if you use them the following information should be helpful. Your printer is probably connected to either a parallel or serial port. MSDOS monitors serial and parallel port activities to detect device hang-ups. While undoubtedly an admirable goal, it can occasionally create irritating ***abort, retry, ignore*** messages on your screen. This can become especially irritating with older PostScript printers (*with slow processors*) that seem to stop and pant for extended periods during output. If you have this problem, include the following ***MODE*** command modifications in your ***AUTOEXEC.BAT*** file:

MODE LPT1,,P	The **P** tells ***MSDOS*** to Persist. ***MSDOS*** will not check for device hang-ups on ***LPT1***.
MODE COM1,96,N,8,1,P	The **P** tells ***MSDOS*** to Persist. ***MSDOS*** will not check for device hang-ups on ***COM1***.

Remember to Initialize Serial Ports

Remember: Serial ports (*in MSDOS*) must be initialized properly via a ***MODE COM?*** command (*as above*) to communicate properly with any printer connected to a serial port. Make sure your ***AUTOEXEC.BAT*** file contains the proper ***MODE*** command if your printer is connected to a serial port.

SIMION and your Windows Printer

SIMION 7.0's GUI ***automatically*** selects the current default Windows printer as its output destination when you first run the program. *Note However*, if you change the output destination to something else (*e.g. Windows clipboard, metafile or GUI PostScript driver*) the selection will be remembered ***between*** SIMION sessions.

The GUI's Printer Pipelines

When you print something from SIMION a switch is set within the GUI and the desired object(*s*) is (*are*) redrawn. During this redraw process each drawing command is automatically routed to ***both*** the Windows screen driver (*redrawing the object(s) on the screen*) ***and*** to the currently active printed output destination (*e.g. the current Windows printer*). ***Thus you can see the printing progress by just looking at your screen!***

Remember: If something is wrong or you want to abort the current printout, just hit the **<ESC>** ***key during this redrawing process.***

The GUI actually has ***two*** printing pipelines: A vector (*floating point*) pipeline and a pixel (*integer*) pipeline. The differences between the two printing strategies are discussed below. However, all printer output whether pixel or vector is ***eventually*** converted to ***either*** integer graphics (*all Windows drivers*) ***or*** floating point graphics (*all GUI printer language drivers*).

Printed images can be positioned and sized on the page as desired via the GUI's print option image composer. The actual quality of the printed (*or stored - clipboard*) image will depend on the quality of the printer ***and its driver***, scaling factor (*a GUI to Windows integer image quality fix*), image size on the page, and whether the output was originally vector or pixel based.

Pixel Verses Vector Printing

The GUI supports ***both*** pixel (*e.g. screen printing*) and vector (*e.g. ion trajectories in View*) printing.

Pixel Printing

Pixel printing can be thought of as transferring copies of your screen's image (*pixels - dots*) to your printer. This means that the printed image shares the resolution and quality of the image you see on your screen. ***All lines are drawn at the equivalent width of one screen pixel.*** Therefore, if SIMION is displayed at a low screen resolution (*800 x 600 pixels*) any pixel printing will be of much lower detail quality than if SIMION is displayed with a high screen resolution (*e.g. 1500 x 1200 pixels*).

Note: Because the GUI ***always*** converts pixel output into vector format (*points converted to line definitions*) before shipping it to your printer (*clipboard or metafile*), the jagged diagonal lines visible on your screen will appear as somewhat smoother diagonal lines on your printer (*the better the printer, the smoother the lines. Also GUI printer language drivers are smoother than Windows drivers*).

Vector Printing

Vector printing involves drawing lines with widths defined in printer space (*points or mm*) rather than in screen space (*pixels*). This means that an isometric view that might look absolutely shabby in low screen resolution generally looks quite acceptable when printed (*floating point based GUI printer language drivers often yield perceptibly better print quality than integer based Windows drivers*).

Where Vector and Pixel Printing are Used

SIMION uses pixel printing for ***all*** ***<u>screen</u>*** printing, and for printing any 2D view within **Modify**. Vector printing is used for isometric views within **Modify** and **<u>all</u>** printing within **View**.

How to Request a Printout

There are two types of program printouts: Window and screen. Each is requested differently.

Requesting a Window Printout

SIMION provides a **Print** button where window printout is supported (*e.g.* **Modify** *and* **View**). Just click the **Print** button to print the window's current view. ***The Annotate and Print Screen will appear (discussed below).***

Requesting a Screen Printout

The entire GUI screen area (*Window's client area*) ***or*** selected screen object(*s*) can be printed via the GUI screen print utility. To access screen printing hold the **<Crtl>** key depressed and press the **<F1>** key. A crosshairs cursor will appear with a panel object in the upper left corner giving its position.

To exit from the screen print utility just hit the <ESC> *key.*

Printing the Entire Screen

To print the entire screen just hit the **<P>** key. You will asked if you want to print the shaded object or the entire GUI screen. Click the **No** button to print the entire GUI screen. The Annotate and Print Screen will appear.

Printing Selected Object(s)

The GUI's screen is composed of a collection of objects. Objects are layered to form parent-child relationships. An object is selected by pointing to it with the cross hairs cursor and pressing the **<P>** key. If you want to want to print a group of objects point to their common parent object (*guess*) and press the **<P>** key.

The GUI will shade the selected object (*not necessarily the one you expected*) and ask if you want to print it. ***If the object is not the one you wanted click the*** **Cancel** ***button and try again.***

When the selected object(s) is (*are*) acceptable press the **Yes** button. The Annotate and Print Screen will appear (*Figure G-1*).

The Annotate and Print Screen

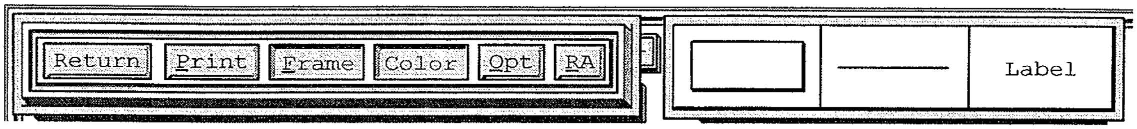


Figure G-1 The Annotate and Print Screen

The Annotate and Print Screen appears when any printout is requested (*Figure G-1 above*). This screen consists of the object to be printed surrounded by a grayed screen with a collection of control buttons and annotation objects in the upper part of the screen.

Print Control Buttons

A row of buttons control access to various printing features.

Return Button

The **Return** button is used to exit the Annotate and Print Screen and its printing functions.

Print Button

Click the **Print** button to initiate the actual printing process to the currently selected output destination device.

Frame Button

The **Frame** button options the drawing of a black shadow frame around the printed object when depressed as shown in Figure G-1.

B/W - Color Button

The **B/W - Color** button selects whether black and white or color output is sent to the printer (*metafile or clipboard*). When the button is depressed it selects **Color** (*Figure G-1*). When raised it displays **B/W** to indicate black and white. The effects of this button is dependent on whether the output destination is a Windows device (*e.g. clipboard*) or a GUI printer language driver.

Windows Printers, Clipboards, and Metafiles

For Windows destination devices, **Color** gives color (*or gray scales on non-color printers*) and **B/W** gives Black and White. Pixel and vector pipelines control line widths as follows:

Pipeline	**Color**	**B/W**
Pixel	Lines One Pixel Wide	Lines One Pixel Wide
Vector	Adjustable Line Widths*	Variable Line Widths

* Color Vector Line Widths = Color 15 (*white's*) Width.

GUI Printer Language Drivers

The effects of the **B/W - Color** button are dependent on the GUI printer language selected (*e.g. PostScript*) and whether the pixel or vector pipeline is being used:

With Pixel Printing:

Language	Color	B/W
HP-GL *	Multi-Pen	Pen 1 Only
HP-GL/2 *	Multi-Pen	Pen 1 Only with Gray Scaling
PCL 5	Color PCL 5	Gray Scale PCL 5
PostScript	Color PostScript	Gray Scale PostScript

* Rectangular Areas Outlined only (*not filled*).

With Vector Printing:

Language	Color	B/W
HP-GL	Multi-Pen	Pen 1 Only
HP-GL/2	Multi-Pen *	Pen 1 Only with Variable Line Widths
PCL 5	Color PCL 5 *	Variable Line Widths
PostScript	Color PostScript *	Variable Line Widths

* Color Vector Line Widths = Color 15 (*white's*) Width.

Note: Line widths are set in the Print Options Screen below.

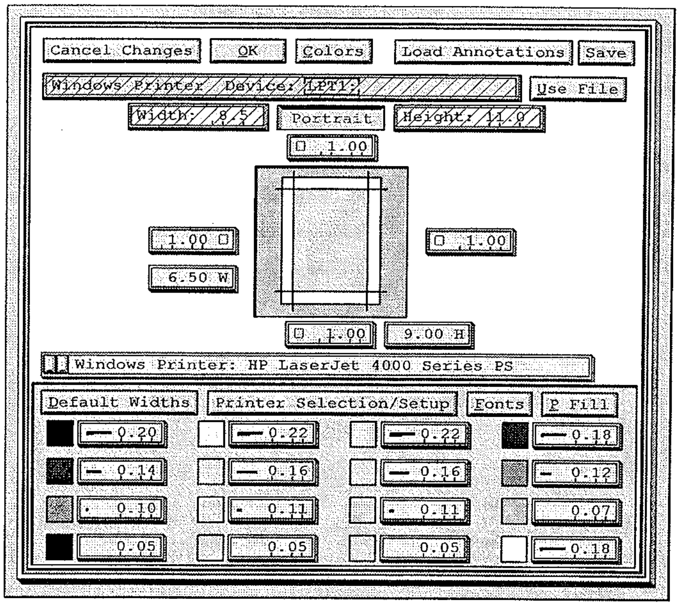
Options Button

The **Options** (*or **Opt***) button accesses the Print Options Screen discussed below.

RA Button (*if visible*)

If you have used any annotations in the previous printout (***even** in a **previous** program session*). The GUI will provide an **RA** button (*Restore Annotations*) to allow you to restore the previous annotations (*very handy*). If you want to restore your previous annotations (*then rip out what you don't want*), be sure to click the **RA** button ***before*** creating any new annotations. ***If you don't, all the previous annotations will be erased!***

You also have the option of saving annotations to **.ANN** files via the Print Options Screen discussed below.



Windows Printer Device: LPT1:

Width: 8.5

Height: 11.0

1.00

1.00

1.00

6.50 W

1.00

9.00 H

Windows Printer: HP LaserJet 4000 Series PS

Figure G-2 The Print Options Screen

Print Options Screen

The Print Options Screen (*Figure G-2*) is used to select the output destination driver, port or file, printer language, page size/margins, colors, line widths to substitute for colors (*B/W Vector outputs*), and more. The GUI saves printer options in the file **C:\FILES.GUI\WPLOTTERS.GUI** so they can be retained between your SIMION sessions. The Print Options Screen is accessed via the **Options** (*or **Opt***) button.

Note: ***Once any option below is changed it will remain the same from session to session.***

Because the Print Options Screen supports Windows printers, clipboards, metafiles, and GUI printer language drivers, there are many options that may be available (*or specified*) for one device that may be unavailable or not specifiable for another (*e.g. number of pens – HPGL*). These issues are resolved by either blanking out options that are not accessible or selectively displaying options only for the devices for which they apply.

The Print Options Screen is divided into three areas. The top of this screen contains a line of control buttons. Just below this line are found the general print options like device/port/file, page size/margins, and page orientation.

Below the general print options area is a selector object that is used to select the destination for printed output (*e.g. Windows clipboard*). In Figure G-2 above, the Windows Printer has been selected, and the default (*or currently selected*) Windows printer driver is the “HP LaserJet 4000 Series PS”.

Immediately below this selector object is the Output Destination Options Screen. The screen displayed in Figure G-2 is the one that supports Windows printers.

The Top Line of Buttons

The functions of the top line of buttons do not change according to printing device selection.

Cancel Changes Button

Clicking the **Cancel Changes** button exits the Print Options Screen restoring the options active when it was entered.

OK Button

Clicking the **OK** button exits the Print Options Screen keeping any options changes.

Colors Button

The Colors button is used to gain access to the GUI's Color Palette Screen. This is handy if you need to change colors to put just the right emphasis on something.

Load Annotations and Save Buttons

The two remaining buttons are used to save the currently defined annotations (*annotations are discussed later*) as an **.ANN** file that can be reloaded later as needed. ***Note: These two buttons do not appear if the GUI File Manager is currently active, because recursive calling of the file manager is not allowed.***

General Print Options

The general print options apply to all printouts. However, their use and meaning may depend on the Windows printer/device or GUI printer language selected.

Printer's Device

This ioline object contains the name of the currently set printing device port or file (*e.g.* **LPT1:** *in Figure G-2 above*). It's function depends on the currently selected output destination.

Windows Printers

For Windows printers the device port is displayed but ***not*** adjustable from within SIMION. If you want Windows printer output sent to a file depress the **Use File** button (*object to the right*) and enter the destination file's name in the ioline.

Windows Clipboard

This ioline object displays that the destination is the clipboard as a reminder.

Windows Metafile

This ioline allows you to enter the name of the Windows metafile to use (*one printout per metafile*). A **.wmf** extension will automatically be appended if necessary.

Windows Enhanced Metafile

This ioline allows you to enter the name of the Windows enhanced metafile to use (*one printout per metafile*). An **.emf** extension will automatically be appended if necessary.

GUI Printer Language Drivers

The GUI printer language drivers (*e.g. PostScript*) allow you to specify ***either*** a printing device port (*e.g.* ***C:\PRN*** *the GUI default*) ***or*** a file name if you want printer output sent to a file. Output is ***always*** appended (*added*) to the device port/file when using GUI printer language drivers. *Note*: For SIMION 6.0 the default device port was **PRN**. However NT requires **C:\PRN** (*and both work with 9x*), so **C:\PRN** is the new SIMION 7.0 default.

Use File Button

The **Use File** button is only accessible when a Windows printer is the selected output device. It directs Windows printer output (*when depressed*) to the file named on the Printer's Device ioline (*discussed above*).

Orientation Button

The Orientation button is used select printer output orientation: Portrait or landscape. This button is accessible for Windows printers and GUI printer language drivers. Access is blocked for Windows clipboards and metafiles, because they are pixel region outputs.

Page Size

Page (*or output*) size is displayed and adjustable according to the output destination selected:

Windows Printers

The page size in inches is displayed but ***not*** adjustable. You must use the **Printer Selection/Setup** button on the Output Destination Options Screen to access the Windows printer driver and change page sizes.

Windows Clipboards, Metafiles, and Enhanced Metafiles

The page size given is the size in ***pixels*** of screen space of the view area or objects to be output. The GUI makes use of a ***scaling factor*** (*with a default value of 10*) to enhance the detail and resolution of these forms of Windows output (*remember Windows is integer graphics – crude*). The Output Destination Options Screen has a **Scaling Factor** panel object to allow you to adjust this factor.

GUI Printer Language Drivers

Page size can be user adjusted to reflect the larger paper size on some printers and drafting plotters. These controls are very useful as tools to trick your printer to work properly or produce some special effect.

Page Margin Controls

Windows Printers and GUI Language Drivers

Four panel objects are provided to define page margins. Your printed output will always be ***sized without distortion*** (*equal x and y scaling*) ***and centered*** to fit within these margins. The ***gray area*** in the page display object ***shows where your output will be placed*** on the page.

Margin settings are reasonably accurate with Windows printers and GUI printer language PostScript. However, PCL 5 devices have differences in their page alignment. ***Therefore, you may have to fiddle with the margins a bit to get images just where you want them.***

Windows Clipboards, Metafiles, and Enhanced Metafiles

Page margins are always zero, because you are creating a metafile image that is to be inserted onto a page or other object.

Output Destination Selector

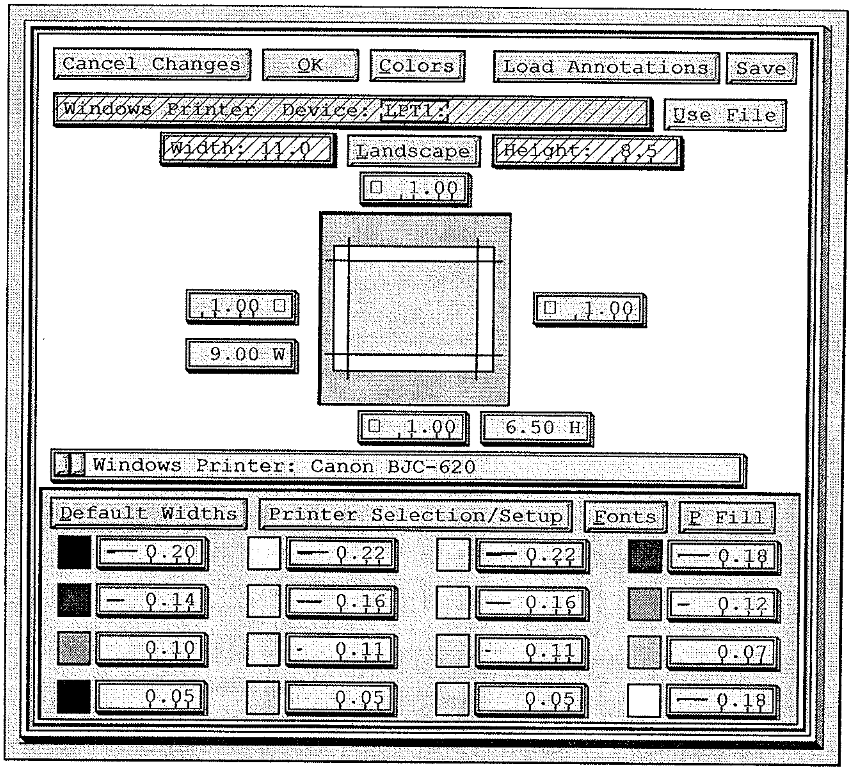
The **Output Destination** selector object is used to select the output's destination (*e.g. Windows Printer in Figure G-2 above*). The options are: Windows printer, clipboard, metafile, enhanced metafile, or a GUI printer language (*e.g. PostScript*). The GUI printer languages supported are: Five PostScript modes along with HP-GL, HP-GL/2, and PCL 5.

When you change between printed output destinations the Output Destination Options Screen (*just below the selector*) will display the specific printer options for the selected destination (*Figure G-3*). The following material discusses what you will see in the Output Destination Options Screen for each output destination:

Currently Selected Windows Printer

Unlike the DOS GUI, the Win32 GUI assumes the default Windows Printer to be the ***<u>default</u>*** ***device.*** When you start SIMION the default Windows printer will be selected as the active Windows Printer (*Figure G-3*).

Printer Selection/Setup Button



Windows Printer Device: LPT1:

Width: 11.0

Height: 8.5

□ 1.00

1.00 □

9.00 W

□ 1.00

□ 1.00

6.50 H

Windows Printer: Canon BJC-620

■ — 0.20	□ — 0.22	□ — 0.22	▨ — 0.18
▨ — 0.14	□ — 0.16	□ — 0.16	▨ - 0.12
▨ 0.10	□ - 0.11	□ - 0.11	▨ 0.07
■ 0.05	□ 0.05	□ 0.05	□ — 0.18

Figure G-3 Windows Printer Destination Options Screen

The Windows printer Output Destination Options Screen contains a **Printer Selection/Setup** button. Click this button to gain access to the standard Windows printer selection and properties screens. Any changes you make will remain active within SIMION (*no impact on other programs*) for the duration of the session.

Output to a Windows Printer generally goes directly/indirectly to the printer via its connection to the computer (*e.g. LPT1*). However, depress the **Use File** button if you want the output to go to a file. Be sure to enter the desired file name (*or use the one suggested*) on the ioline object to the left. The **.prn** file name extension is applied to any file name supplied without an extension.

Fonts Button

All four Windows devices (*Windows Printer, Clipboard, Metafile, and Enhanced Metafile*) have a **Fonts** button on their Output Destination Options Screens. The **Fonts** button is used to access the Windows font selection screen. The font displayed is the currently active default ***TrueType*** font to use with all annotation labels (*discussed under* *annotations below*). When you change the default font all annotation labels that ***do not*** have specific font override definitions will automatically track the change.

P Fill Button

All four Windows devices (*Windows Printer, Clipboard, Metafile, and Enhanced Metafile*) have a **P Fill** button on their Output Destination Options Screens. **P Fill** stands for **polygon fill**. When this button is depressed each line segment is converted to a closed shaded polygon when output to a Windows device.

Unfortunately the Windows GDI appears to have been created by sadistic forces. It turns out that there are ***pure*** colors and ***dithered*** colors. Lines are always drawn in pure colors. Areas are always shaded in dithered colors. The only time that pure and dithered colors are the same is when the device is pure rgb (*e.g. Postscript or a 16bit or above color display*) or when a color palette exists to define a pure color for any that might have to be *dithered (the trick used to make the GUI screen colors display properly with both lines and shaded areas*). Thus, if you are going to output to a lesser printer than PostScript (*e.g. a PCL5 printer: LaserJet 4*) the color (*or gray scale*) of the lines may ***not*** match the color of shaded areas. When the **P Fill** button is depressed lines are converted to shaded (*dithered color*) polygons and things look better. The precision of the output is less, but at least the colors and gray scales are done properly.

If you output to a PostScript or other quality rgb printer (*directly or indirectly*), you will get the best quality and speed if the **P Fill** button is ***not*** depressed (*not active*). If the ultimate disposition of the output is uncertain (*e.g. a clipboard image*), it is generally safer to have the **P Fill** button depressed (*active*).

The status of the **P Fill** button is ***remembered*** between SIMION sessions.

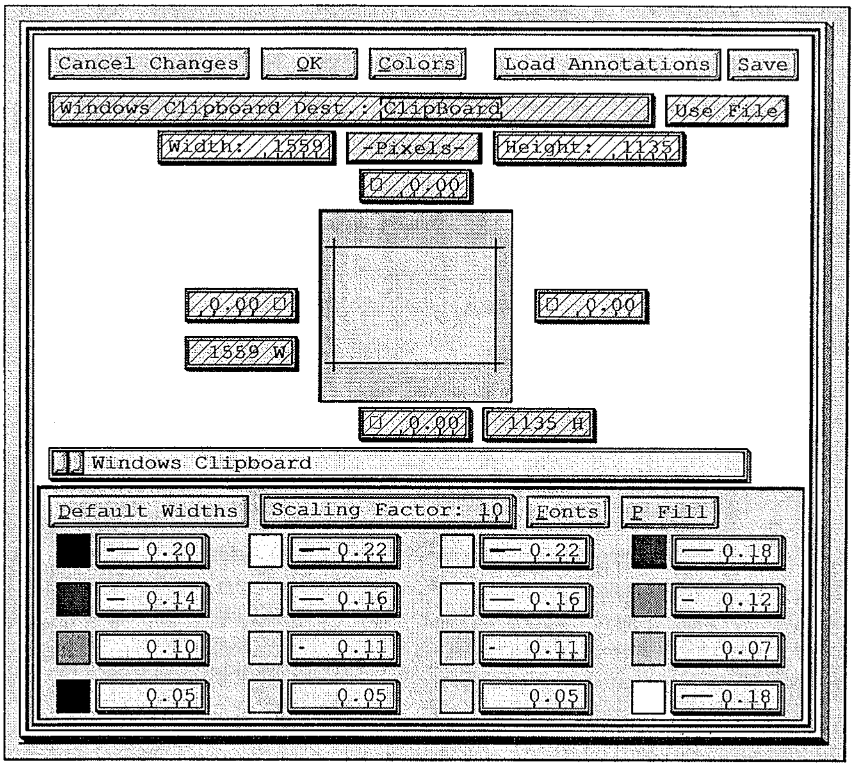
Default Widths Button

All four Windows devices (*Windows Printer, Clipboard, Metafile, and Enhanced Metafile*) have a **Default Widths** button on their Output Destination Options Screens. The **Default Widths** button restores the default translation of line color to line widths for B/W output through the ***vector*** pipeline (*e.g. printing ion trajectories in View*).

Color to Line Width Conversion Panels

All four Windows devices (*Windows Printer, Clipboard, Metafile, and Enhanced Metafile*) have 16 Color to **Line Width Conversion** panel objects on their Output Destination Options Screens. These panels convert a GUI color index (*one of 16 GUI adjustable colors*) into a line width in mm. Colors are translated into line widths for B/W output through the ***vector*** pipeline. ***Changes to a line width are remembered between SIMION sessions.*** Windows line widths only affect the four Windows devices above. The GUI printer language drivers have their own line width tables.

Note: The width of the color white (*15 - last color*) is used to set the width of all vector lines drawn in Color. Thus if you want to simulate fill or other effects in color vector output, change the line width for white.



Windows Clipboard Dest.: ClipBoard

Width: 1559 Height: 1135

0.00 0.00 0.00 0.00

1559 W 1135 H

Windows Clipboard

Scaling Factor: 10

■	— 0.20	□	— 0.22	□	— 0.22	■	— 0.18
■	— 0.14	□	— 0.16	□	— 0.16	▨	- 0.12
▨	0.10	□	- 0.11	▨	- 0.11	▨	0.07
■	0.05	▨	0.05	□	0.05	□	— 0.18

Figure G-4 Windows Clipboard Destination Options Screen

Windows Clipboard

When the Windows Clipboard is the output destination, a ***metafile*** of the selected output is created and stored on the clipboard for use by other programs (*Figure G-4*). This output is pixel based so its resolution is based on the pixel size (*resolution*) of the printed object on your screen. The **Scaling Factor** panel object has been provided to improve the integer image quality.

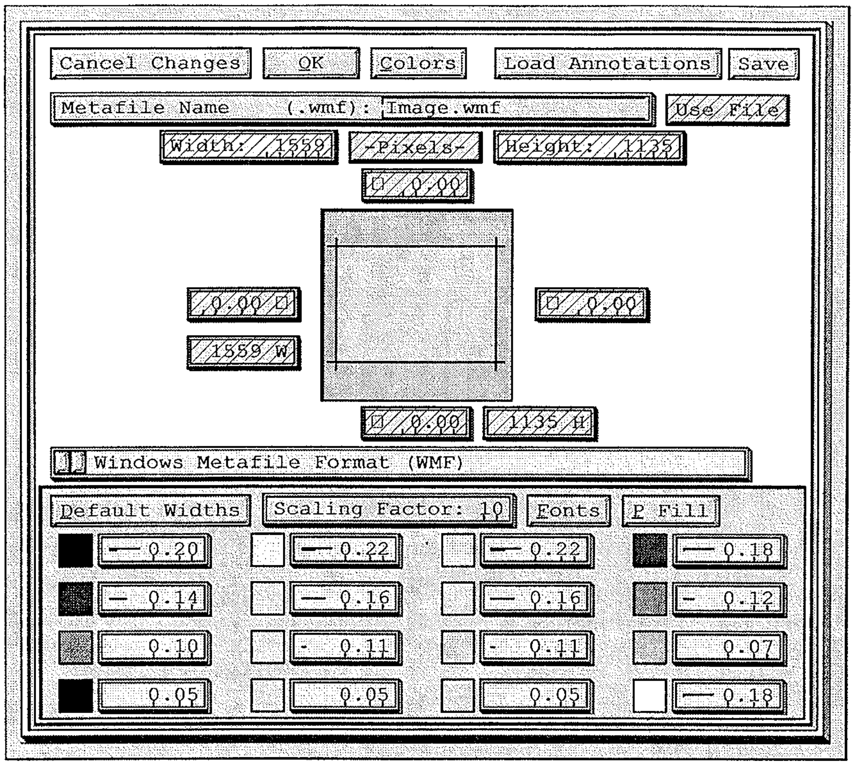
The Scaling Factor Panel

The **Scaling Factor** panel object appears on the Clipboard, Metafile, and Enhanced Metafile Output Destination Options Screens. This panel is used to increase the resolution of the output by a scaling *factor (the maximum **and default** value is **10**)*. When the scaling factor is 10 the destination device *(e.g. Clipboard)* is assumed to have 10 times the pixel resolution of the screen, and therefore higher quality output than you actually see on the screen (*useful*). This sort of thing is necessary to help improve image quality because Windows GDI is only integer based while the GUI supports floating point graphics too.

Note: The Scaling Factor value must be greater than one or polygon filling will not be used (if optioned) when outputting to Clipboards, Metafiles, or Enhanced Metafiles.

Fonts Button, P Fill Button, Default Widths Button, and Color to Line Width Panels

These button and panel objects have exactly the same function and access the same parameters as described for the Windows Printers Destination Options Screen above.



Metafile Name (.wmf): Image.wmf

Width: 1559 -Pixels- Height: 1135

□ 0.00

0.00 □

1559 W

□ 0.00

□ 0.00

1135 H

Windows Metafile Format (WMF)

Scaling Factor: 10

■ — 0.20	□ — 0.22	□ — 0.22	▨ — 0.18
▨ — 0.14	□ — 0.16	□ — 0.16	▨ - 0.12
▨ 0.10	□ - 0.11	□ - 0.11	□ 0.07
■ 0.05	□ 0.05	□ 0.05	□ — 0.18

Figure G-5 Windows Metafile Destination Options Screen

Windows Metafile

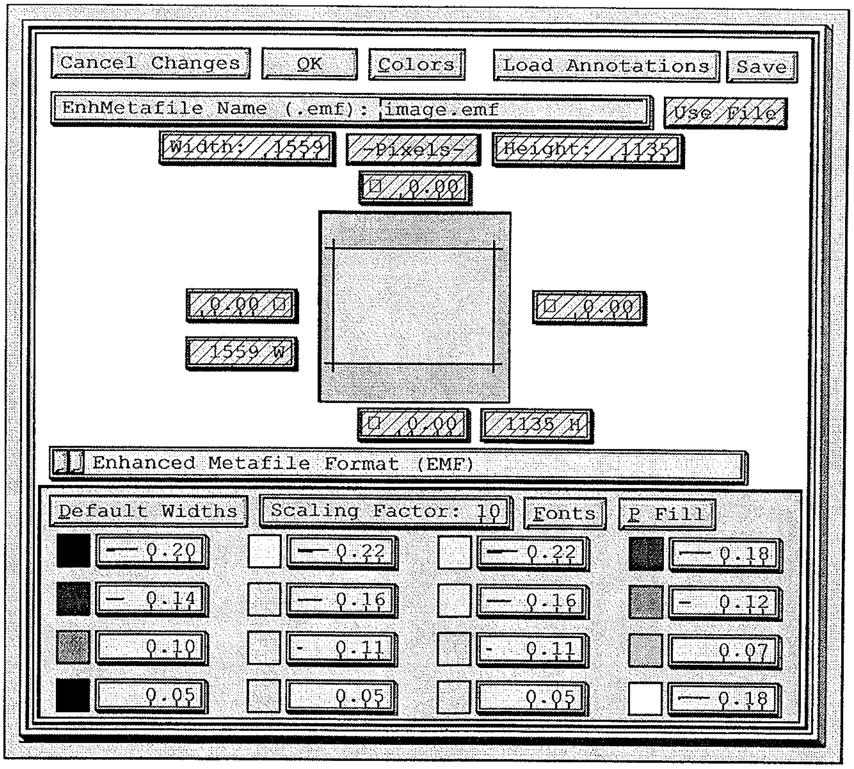
When the Windows Metafile is the output destination (*Figure G-5*), a metafile of the selected output is created and stored to the output file name designated (*one file per printed output*). If an extension is not supplied the **.wmf** extension will be applied. This output is pixel based so its resolution is based on the pixel size (*resolution*) of the printed object on your screen. The **Scaling Factor** panel object has been provided to improve the integer image quality.

The Scaling Factor Panel

The **Scaling Factor** panel object has the same function and accesses the same parameter as described for the Clipboard Destination Options Screen above.

Fonts Button, P Fill Button, Default Widths Button, and Color to Line Width Panels

These button and panel objects have exactly the same function and access the same parameters as described for the Windows Printer Destination Options Screen above.



EnhMetafile Name (.emf): image.emf

Width: 1559 | Height: 1135

0.00 | 0.00 | 0.00 | 0.00

1559 W | 1135 H

□ Enhanced Metafile Format (EMF)

Scaling Factor: 10

■	— 0.20	□	— 0.22	□	— 0.22	■	— 0.18
■	— 0.14	□	— 0.16	□	— 0.16	▨	- 0.12
▨	0.10	□	- 0.11	□	- 0.11	▨	0.07
■	0.05	□	0.05	□	0.05	□	— 0.18

Figure G-6 Windows Enhanced Metafile Destination Options Screen

Windows Enhanced Metafile

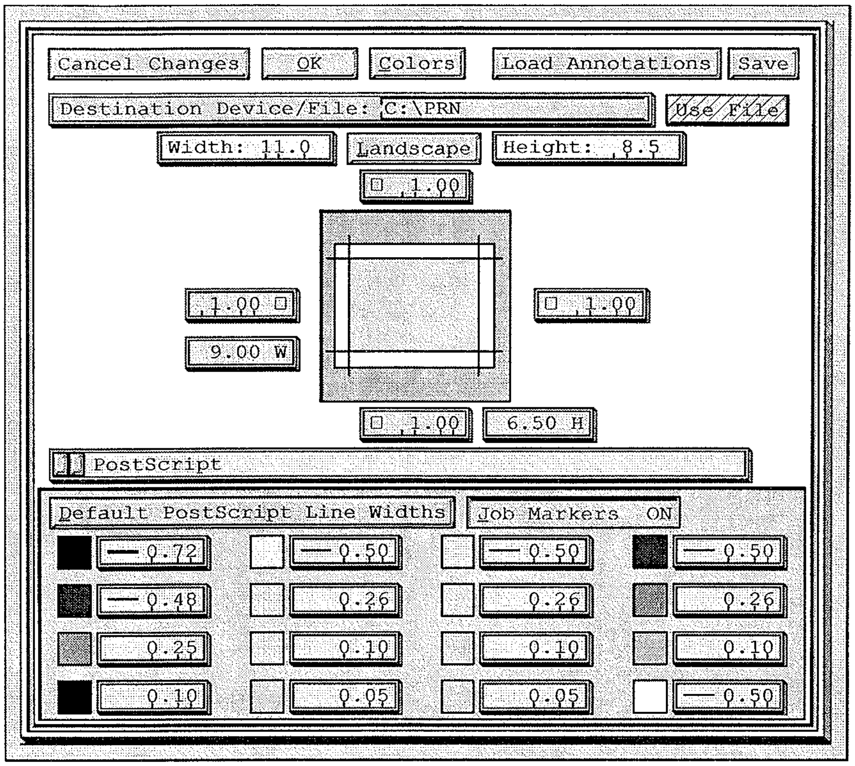
When the Windows Enhanced Metafile is the output destination (*Figure G-6*), an enhanced metafile of the selected output is created and stored to the output file name designated (*one file per printed output*). If an extension is not supplied the **.emf** extension will be applied. This output is pixel based so its resolution is based on the pixel size (*resolution*) of the printed object on your screen. The **Scaling Factor** panel object has been provided to improve the integer image quality.

The Scaling Factor Panel

The **Scaling Factor** panel object has the same function and accesses the same parameter as described for the Clipboard Destination Options Screen above.

Fonts Button, P Fill Button, Default Widths Button, and Color to Line Widths Panels

These button and panel objects have exactly the same function and access the same parameters as described for the Windows Printers Destination Options Screen above.



Destination Device/File: C:\PRN

Width: 11.0

Height: 8.5

□ 1.00

1.00 □

□ 1.00

9.00 W

□ 1.00

6.50 H

PostScript

Job Markers ON

■	— 0.72	□	— 0.50	□	— 0.50	■	— 0.50
■	— 0.48	□	0.26	□	0.26	■	0.26
■	0.25	□	0.10	□	0.10	■	0.10
■	0.10	□	0.05	□	0.05	□	— 0.50

Figure G-7 GUI PostScript Printer Language Options Screen

GUI PostScript Printer Language Driver Modes

The GUI PostScript printer language driver has five modes: ***PostScript, PS 1st Image, PS Next Image, PS Last Image, and PS Encapsulated***. The following describes when and how to use each PostScript mode:

PostScript

This is the normal **one image per page** mode (*the one you should normally use Figure G-7*).

PS 1st Image, PS Next Image, and PS Last Image

These three modes are provided to allow you to create multiple images on a single page. Adjust the margins ***before*** printing each image to position each in the desired page location.

PostScript uses an Opaque printing model. This means that ***later printed images cover earlier printed images*** in any areas of overlap.

It is recommended that multiple image compositions be ***output to a file***, and then copied to the printer. ***Reason:*** Most ***PostScript printers*** will automatically ***time-out*** while you're fiddling with your composition (*very irritating*).

PS Encapsulated

This mode is provided to permit you to ***export illustrations*** for inclusion in your documents. ***Save one image per file***. Most word processors expect a file extension (*e.g. .EPS*). ***Suggestion: Zero your margins to maximize your image size.***

Note: This mode will not work properly if sent directly to your printer. It needs to be encapsulated into some other PostScript output stream.

GUI PostScript Printer Language Options Screen

PostScript modes support the following options (*Figure G-7*):

Job Markers

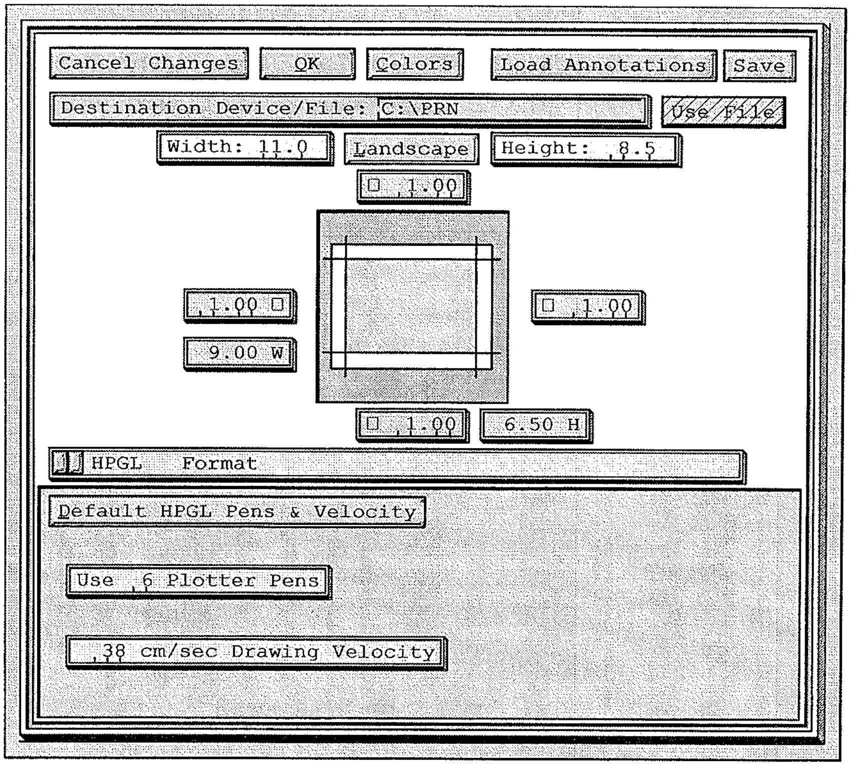
The ***Job Marker*** button is provide to include a **<Ctrl D>** at the beginning and ending of each PostScript page. Most PC based PostScript devices expect to see this marker. However, other systems do not like it (*e.g. Mac*). Use whatever flavor works for you.

Note: PS Encapsulated ignores this option.

Line Width Options

When ***vector*** graphics (*e.g. ion trajectories in View*) are output in B/W with a PostScript driver, colors are converted into variable width black lines. Each color has an associated line width in ***points***. Panel objects are provided to allow you to adjust the widths of each color's line. The default line width settings provide the visual effects of illumination. However, they can be adjusted as desired.

Note: The width of the color white (*15 - last color*) is used to set the width of all vector lines drawn in Color PostScript. Thus if you want to simulate fill or other effects in color vector output, change the line width for white.



Destination Device/File: C:\PRN

Width: 11.0

Height: 8.5

□ 1.00

1.00 □

□ 1.00

9.00 W

□ 1.00

6.50 H

HPGL Format

Use 6 Plotter Pens

38 cm/sec Drawing Velocity

Figure G-8 GUI HP-GL Printer Language Options Screen

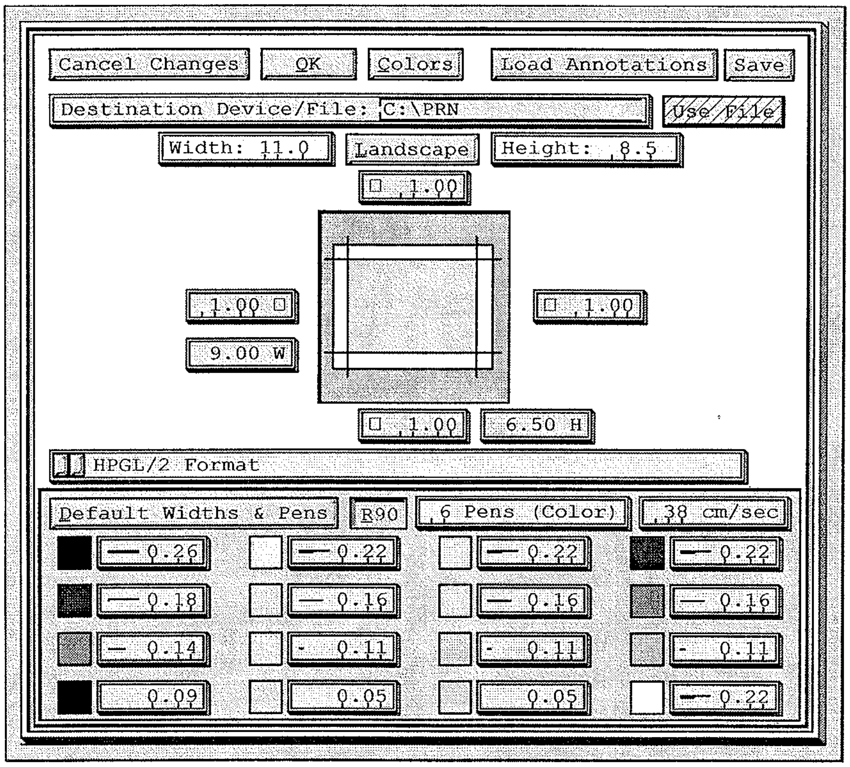
GUI Printer Drivers for HP-GL, HP-GL/2, and PCL 5

These ***three drivers are all really HP-GL(/2) based*** with different assumptions concerning the target printing device (HP-GL Figure G-8). ***Therefore, all three drivers can be used with an HP 7475A plotter***. The error light may flash like crazy but a plot will emerge. Likewise, all three drivers will generate (*some form of*) output on a HP-GL/2 or PCL 5 device. The primary limitation is that for the HP-GL driver to work properly the printer must be in HP-GL/2 mode (*the **ESC%-1B** has been transmitted to the PCL 5 Printer*).

Both **HP-GL and HP-GL/2 drivers assume** the device is actually a **pen plotter**. Thus they **support** options such as specifying the **number of pens** and the setting of **drawing velocity** (for transparencies with pen based plotters). **B/W drawing** is **always** done using **pen 1**. Neither fills rectangular areas in pixel drawing (*because a transparency model is assumed*). **HP-GL/2** assumes the device can **support variable line widths**, **gray scaling**, and also provides **page size definitions** and cutting needed by drafting plotters with a roll type paper source.

PCL 5 assumes that the device is a fully capable HP-GL/2 device (*job stream controllable colors/grays and line widths*). **Note: Pen zero is considered to be an active color (*black*).**

Thus if one driver doesn't quite work try another. There are enough options to trick most HP compatible devices into working.



Destination Device/File: C:\PRN

Width: 11.0

Height: 8.5

□ 1.00

1.00 □

□ 1.00

9.00 W

□ 1.00

6.50 H

HPGL/2 Format

6 Pens (Color)

38 cm/sec

■ — 0.26 □ — 0.22 □ — 0.22 ▩ — 0.22

▩ — 0.18 □ — 0.16 □ — 0.16 ▨ — 0.16

▨ — 0.14 □ - 0.11 □ - 0.11 ▨ - 0.11

■ 0.09 □ 0.05 □ 0.05 □ — 0.22

Figure G-9 GUI HP-GL/2 Printer Language Options Screen

R90 Option Button (HP-GL/2 & PCL 5)

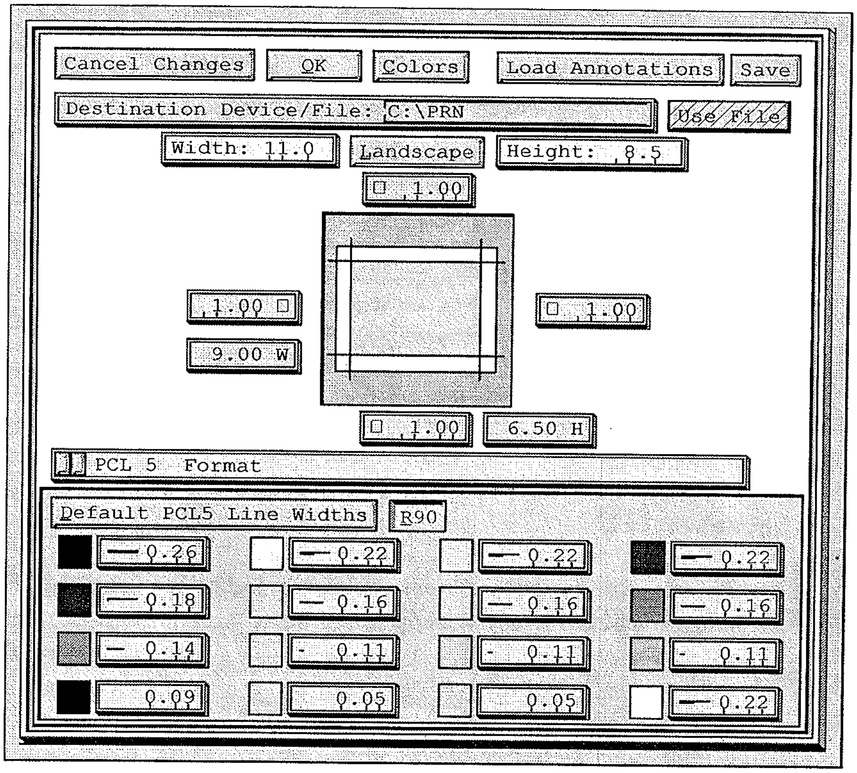
When this button is depressed the image is rotated 90 degrees from HP-GL/2 standards (*Figures G-9 and G-10*). This button is normally depressed for HP-GL/2 and PCL 5 output to LaserJets (*raised for DeskJets*). ***If your plot is twisted 90 degrees and clipped off try the opposite R90 option***.

Number of Pens (HP-GL and HP-GL/2)

This panel object sets the maximum number of pens (*from 1-16 Figures G-8 and G-9*). Set the number to that corresponding to your plotter ***or*** the maximum number of pens you actually want to use.

Drawing Velocity (HP-GL and HP-GL/2)

The drawing velocity panel object sets the pen drawing speed in *cm/sec* (*Figures G-8 and G-9*). This is useful for pen plotters to control drawing quality on media like transparencies.



Destination Device/File: C:\PRN

Width: 11.0

Height: 8.5

□ 1.00

1.00 □

9.00 W

□ 1.00

□ 1.00

6.50 H

PCL 5 Format

0.26	0.22	0.22	0.22
0.18	0.16	0.16	0.16
0.14	0.11	0.11	0.11
0.09	0.05	0.05	0.22

Figure G-10 GUI PCL 5 Printer Language Options Screen

Line Width Options (HP-GL/2 and PCL 5)

When ***vector graphics*** is output in B/W with an HP-GL/2 or PCL 5 driver, colors are converted into variable width black lines (*Figures G-9 and G-10*). Each color has an associated line width in mm. Panel objects are provided to allow you to adjust the widths of each color's line. The default line width settings provide the visual effects of illumination. However, they can be adjusted as desired.

Note: The width of the color white (15 - last color) is used to set the width of all vector lines drawn in Color HP-GL/2 and PCL 5. Thus if you want to simulate fill or other effects in color vector output, change the line width for white.

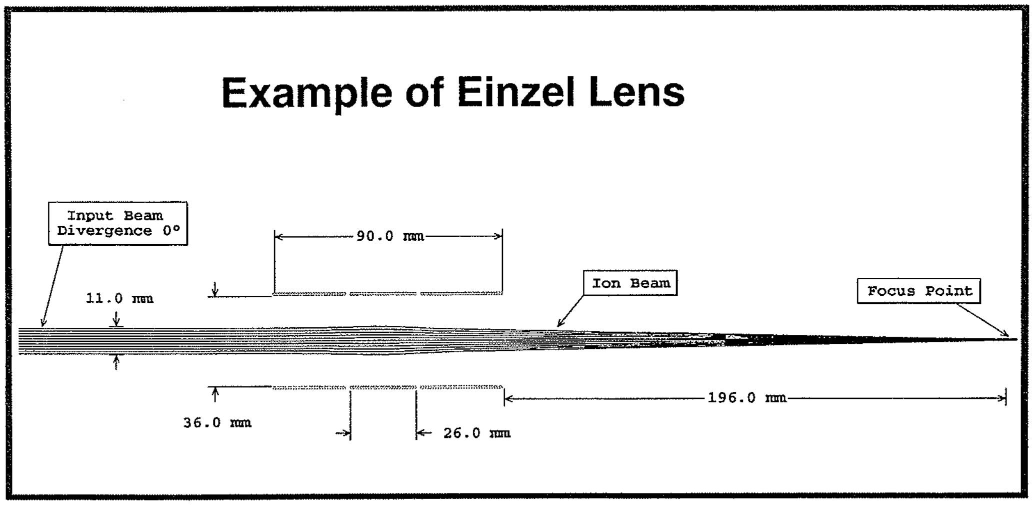


Figure G-11 Example of an annotated output

Annotation Options - Henry Ford: "You can have any color as long as it's black"

All printed outputs from SIMION can make use of the GUI's annotator (*Figure G-11*). The annotator provides an easy way to add legends, labels, boxes, dimensions, arrows and lines to mark or clarify your illustration. It is not intended to serve all needs. ***However, it is quick, easy to learn, and well suited to most annotation tasks.***

If you choose to make use of the annotator be sure to annotate ***before*** you actually print. When a view is annotated (*has annotation objects on it*), the GUI will always return to this view after each printout (*instead of exiting the print menu*). ***This allows you the option of further editing your annotations without having to recreate them between printouts.***

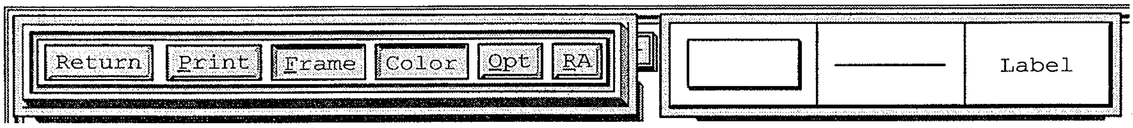


Figure G-12 The Annotate and Print Screen

Moving the Print and Annotate Screen

The Print and Annotate Screen is normally at the top of your screen (*Figure G-12*). In some cases it may block an area you want to annotate. To toggle the Print and Annotate Screen to the bottom of your screen place the cursor in the view area (*but not over any annotation object*) and press the **<spacebar>** key. Hitting the **<spacebar>** key again will toggle the Print and Annotate Screen back to the top of the screen.

Annotation Objects

There are three classes of annotation objects: ***Boxes, lines, and labels***. Each class is represented by a sample object drawing of itself at the top of your screen (*Figure G-12*).

Each of the three types of annotation objects (*boxes, lines, and labels*) have greatly expanded capabilities in SIMION 7.0. Boxes can blank areas without displaying boundaries. Moreover, they can be used to display dimension structures. Lines can be extended, have arrows on either or both ends, and have their length be used to create dimension labels. Labels can optionally blank the area below them (*helps erase dimension lines*). Moreover, when a Windows device (*printer, metafile, or clipboard*) is the output device, you have full access to Windows TrueType fonts including the ability to use separate fonts and sizes for each label note.

Access to these expanded annotation object features involves pointing to the desired annotation object (*e.g. a label*) and clicking a Ctrl key to access each option. ***For a help screen on what options exist and their access keys for an annotation object, point the cursor to it and press the F1 key.***

The Importance of the Order of Creation for Objects

All objects are printed in the order of their creation (*oldest - first, youngest - last*). This is normally not a problem except for annotate boxes (*and dimensions*). Because the printing model is ***opaque*** (*objects can't be seen through objects*), a annotate box object created ***after*** the label it contains ***will cover*** (*obscure*) the label when it is printed. If this happens point to the offending box/dimension object and hit ***any*** *<arrow>* key. ***This will lower the box object to the bottom of the display chain.***

Basic Operations on Single Annotation Objects

The following basic operations are performed the same way for all single (*ungrouped*) annotation objects:

Creating an Annotation Object

An annotation object is created by simply pointing the cursor to the class of object you want (*Figure G-12*) and dragging a copy of it (*with* **either** *mouse button depressed*) to the desired location on the view. ***A maximum of 200 annotation objects can exist at one time.***

Moving an Annotation Object

To move a previously created annotation object point the cursor to it and ***drag it using the* right *mouse button.***

Deleting an Annotation Object

An annotation object is deleted if is **dragged out of the view** (*into the gray area or off the screen*).

Copying an Annotation Object

To obtain a copy of an annotation objects point to it with the cursor, ***hold down the* <Ctrl> *key*** and drag off a copy with the **right** mouse button depressed.

Operations with Groups of Annotation Objects

Annotation objects can be marked and operated on as if they were part of the same group. This feature allows ***moving***, ***copying***, and ***deleting*** annotation objects ***in marked groups***.

Marking a Group of Objects

A group of objects is marked by dragging a ***rectangle to completely surround*** the desired grouping (*with* **either** *mouse button depressed*). The starting point should be outside of any actual object as should be the stopping point.

If no objects exist within the marked area the computer will beep and the rectangle will erase when the mouse button is released.

Moving a Marked Group of Objects

A marked group of objects can be moved by dragging it with ***either mouse button depressed.***

Deleting a Marked Group of Objects

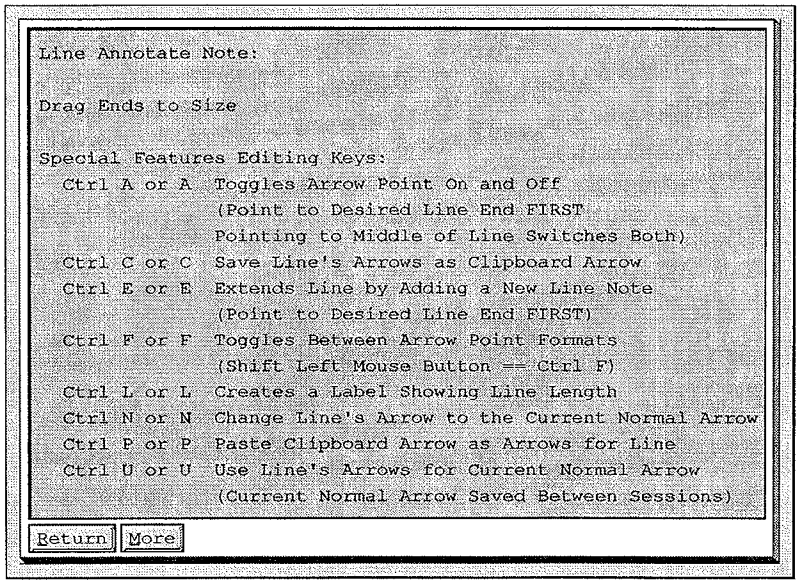
A marked group of objects may be deleted by dragging it into the gray area outside the view or off the page.

Copying a Marked Group of Objects

Once a group of objects has been marked it can be copied by ***holding down the* <Ctrl> *key*** and ***then*** dragging off a new copy of the group with ***either mouse button depressed.***

Editing Individual Annotation Objects

Each class of annotation object can be edited. ***Editing is generally done by dragging parts of the annotation object with the* left *mouse button and/or entering keystrokes.*** Each annotation object has a summary help screen on its editing options. ***Just point to the object you want to edit and hit the <F1> key.***



```
Line Annotate Note:

Drag Ends to Size

Special Features Editing Keys:
  Ctrl A or A  Toggles Arrow Point On and Off
               (Point to Desired Line End FIRST
               Pointing to Middle of Line Switches Both)
  Ctrl C or C  Save Line's Arrows as Clipboard Arrow
  Ctrl E or E  Extends Line by Adding a New Line Note
               (Point to Desired Line End FIRST)
  Ctrl F or F  Toggles Between Arrow Point Formats
               (Shift Left Mouse Button == Ctrl F)
  Ctrl L or L  Creates a Label Showing Line Length
  Ctrl N or N  Change Line's Arrow to the Current Normal Arrow
  Ctrl P or P  Paste Clipboard Arrow as Arrows for Line
  Ctrl U or U  Use Line's Arrows for Current Normal Arrow
               (Current Normal Arrow Saved Between Sessions)
```

Figure G-13 Help Screen for Annotate Line Objects

Editing Line Objects

The capabilities of annotation lines have been expanded: Arrow heads, connected lines, line length labels and etc. ***For a summary help screen (Figure G-13), point to an annotation line object and hit the <F1> key.***

Stretching a Line Object

A line object is stretched by pointing the cursor to one of its ***end points*** (*a small square will surround the end point*), and then dragging that end point with the **left *mouse button depressed.***

Annotation Lines can Show Their Length

When you point to the ***end*** or ***stretch*** an annotation line, the line's length ***may*** be displayed. Display of line lengths is limited to those views in which the Where function (*if active*) would show coordinates on the cursor (*e.g. Modify and View. Note: Where doesn't actually have to be active.*). The units will be the same as Where would show. ***The lengths are corrected for any current asymmetrical scaling in View.***

This feature is handy if you want to include scales. Just stretch a annotation line to the desired scale length (*using the length display*). Now position it as desired. You may want to use the annotation line length label feature discussed below to add a dimension label.

Keyboard Editing of Annotation Lines .

You have the option of extending a line with new lines, adding arrowheads to the end of an annotation line, and creating annotation length labels for it via keystroke editing.

Ctrl A or A Toggles Arrowhead(*s*) on and off. To toggle a single arrowhead, point to the end of the line and hit the **A** key (*or Ctrl A*). To toggle both arrowheads, point to the middle of the line and hit **A**.

Ctrl C or C Copies (*saves*) the arrowhead format of the selected annotation line to the arrowhead format clipboard. This clipboard is ***local*** to the program. The **Ctrl P** function **P**astes this copied arrowhead format on selected lines and dimension structures (*boxes*).

Ctrl E or E Extends line by adding new annotate line to its end. Point to end of a line, hit **E,** and drag the end of the new line segment to the desired location.

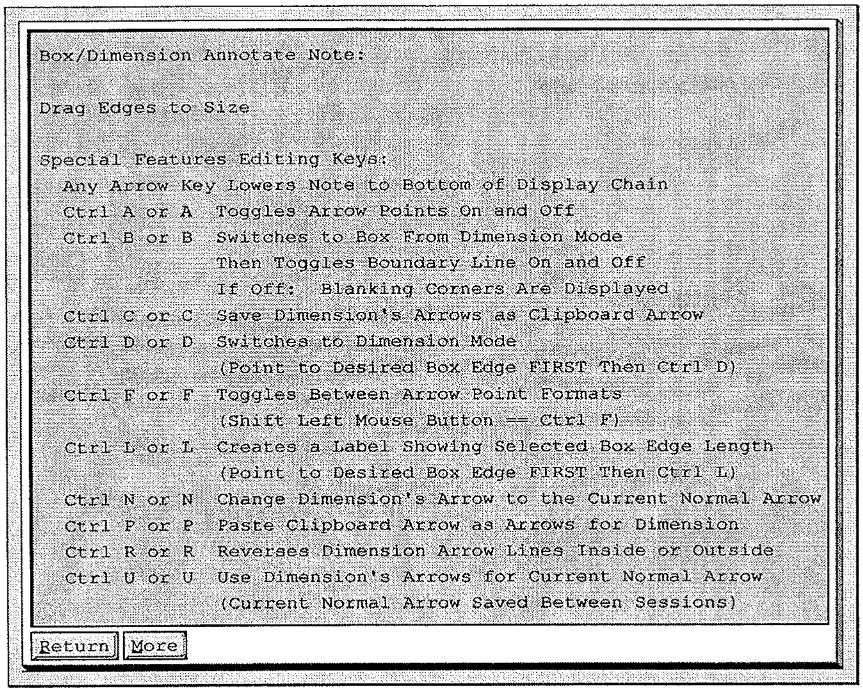
Ctrl F or F Toggles between the available arrowhead Formats. Individual annotation lines can have different arrowhead formats. Note: All annotation lines are created to automatically track the ***current normal arrowhead format*** (*See **Ctrl N** and **Ctrl U** for details*). Note: **Shift Left Mouse** button is same as **Ctrl F** or **F**.

Ctrl L or L Creates a blanked area annotation Label that displays the Length of the selected line. This annotation label can be edited and moved about in the normal way. The value shown for the dimension varies. By default it is the length of the line in pixels. However, in Modify and View it will be in the units that Where would currently show (*if it were active*). This only works in 2D views. Grid units in View require that the currently selected instance is ***integrally*** aligned with the workbench coordinates (*use the Align button as required*).. ***The dimensions are corrected for any current asymmetrical scaling in View.***

Ctrl N or N Changes the arrowhead format of the selected annotation line to automatically track the ***current Normal arrowhead format.*** All new annotation lines and dimension structures (*from boxes*) are set to automatically track the ***current normal arrowhead format***. The ***current normal arrowhead format*** is ***retained*** between program sessions.

Ctrl P or P Pastes the arrowhead format stored in the arrowhead format clipboard (*by Ctrl C*) to the currently selected annotation line.

Ctrl U or U <u>U</u>ses the selected annotation line's arrowhead format for the ***new current normal arrowhead format,*** and restores the selected line into tracking the ***current normal arrowhead format***. The ***current normal arrowhead format*** is used for all new annotation lines and dimension structures (*boxes*) and is saved between program sessions.



```
Box/Dimension Annotate Note:

Drag Edges to Size

Special Features Editing Keys:
  Any Arrow Key Lowers Note to Bottom of Display Chain
  Ctrl A or A  Toggles Arrow Points On and Off
  Ctrl B or B  Switches to Box From Dimension Mode
               Then Toggles Boundary Line On and Off
               If Off:  Blanking Corners Are Displayed
  Ctrl C or C  Save Dimension's Arrows as Clipboard Arrow
  Ctrl D or D  Switches to Dimension Mode
               (Point to Desired Box Edge FIRST Then Ctrl D)
  Ctrl F or F  Toggles Between Arrow Point Formats
               (Shift Left Mouse Button == Ctrl F)
  Ctrl L or L  Creates a Label Showing Selected Box Edge Length
               (Point to Desired Box Edge FIRST Then Ctrl L)
  Ctrl N or N  Change Dimension's Arrow to the Current Normal Arrow
  Ctrl P or P  Paste Clipboard Arrow as Arrows for Dimension
  Ctrl R or R  Reverses Dimension Arrow Lines Inside or Outside
  Ctrl U or U  Use Dimension's Arrows for Current Normal Arrow
               (Current Normal Arrow Saved Between Sessions)
```

Figure G-14 Help Screen for Annotate Box/Dimension Objects

Editing Annotation Box Objects

Annotation boxes have greatly expanded capabilities, including area blanking and dimension structures. Annotation boxes can exist as ***either*** a box ***or*** as a dimension structure, and can be switched at will between these two structures. ***For a summary help screen (Figure G-14), point to an annotation box and hit the <F1> key***.

Annotation Boxes/Dimensions can NOW be Lowered

Annotation box or dimension structures can block the view of a label if they are created **after** the label. This has been fixed by allowing you to lower a box/dimension to the bottom of the display *chain (displayed below all annotation objects - first)*. To lower a box or dimension structure, point to it with the cursor and hit ***any*** ***<u>arrow</u>*** key.

Sizing an Annotation Box (or Dimension)

An annotation box/dimension object is sized by pointing the cursor to one of its four ***side lines*** (*a small square will surround the side line*), and then dragging that side line with the **left** ***mouse button depressed*** to size the legend box.

Annotation Box as a Box Structure

The following keystroke command supports the box structure:

Ctrl B or B Switches a dimension structure ***into*** a <u>B</u>ox structure. If it is already a box structure, it ***toggles*** a box structure ***between*** a <u>B</u>ox and <u>B</u>lanking area. A blanking area is a box without any borders drawn. When the blanking area mode is active, its corner points will be displayed (*but not printed*).

Annotation Box as a Dimension Structure

Dimension structures are created using the box location and dimensions. The actual dimension arrow line(*s*) can lie along any one of the ***four*** sides of the box. The perpendicular dimension serves to control the length of the construction lines that the dimension arrow line(*s*) point to. Dimension arrow lines can be inside or outside. Arrowhead format can be selected as with *lines (including no arrowheads)*. Dimension structures can be moved and sized in the same manner box structures are moved and sized. The following function creates the dimension structure:

Ctrl D or D Switches a box structure to a Dimension structure (*or changes the side for its dimension arrow lines*). Point to the side of the box (*or dimension structure*) you want the dimension arrow lines on (*edge stretch cursor must be visible*) and then hit the **D** key.

An annotation box structure will be transformed into a dimension structure with arrows active and will automatically track ***the current normal arrowhead format***. If the object is ***already*** a dimension structure, only the side for its dimension arrow lines will be changed (*arrow line type and arrow format will be retained*).

The following functions are used to customize a dimension structure:

Ctrl A or A Toggles dimension structure Arrowheads ***on*** and ***off***.

Ctrl C or C Copies (saves) the arrowhead format of the selected dimension structure to the arrowhead format clipboard. This clipboard is local to the program. The **Ctrl P** function pastes this copied arrowhead format on selected lines and dimension structures (*boxes*).

Ctrl F or F Toggles between the available arrowhead Formats. Individual dimension structures can have different arrowhead formats. Note: All dimension structures are created to automatically track the ***current normal arrowhead format*** (*See Ctrl N and Ctrl U for details*). Note: **Shift Left Mouse** button is same as **Ctrl F** or **F**.

Ctrl L or L Creates a ***blanked*** area annotation Label that displays the Length of a dimension structure edge. Note the **L** keystroke function will create a length label that shows the length of the selected side of the box or dimension structure. Thus it is ***important*** that you point to the edge you want the length of (*edge cursor visible*) and then hit the **L** key.

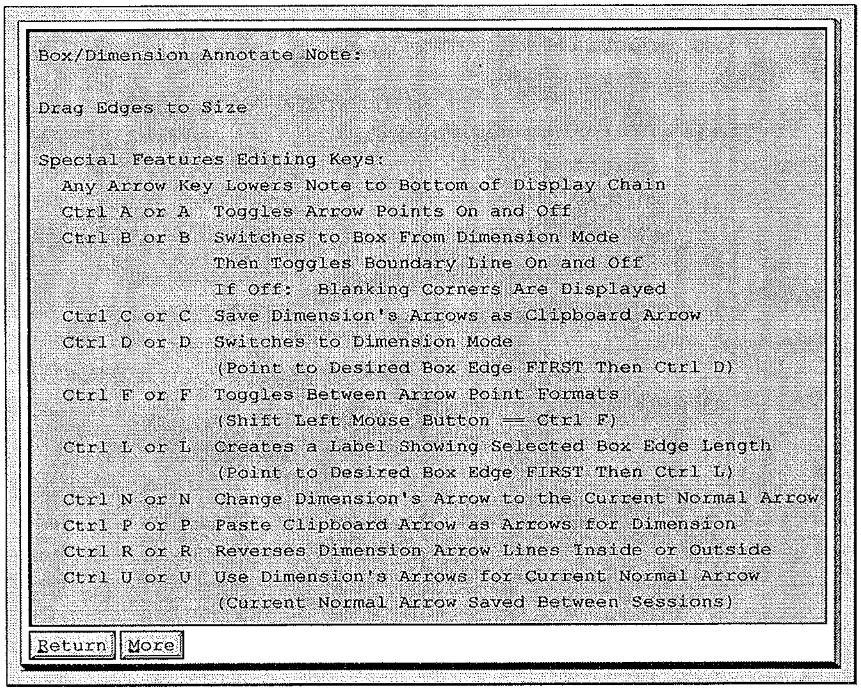
This annotation label can be edited and moved about in the normal way to place it where you want it on the dimension structure. The value shown for the length varies. By default it is the length in pixels. However, in Modify and View it will be in the units that **Where** would currently show (*if it were active*). This only works in 2D views. Grid units in View require that the currently selected instance is integrally aligned with the workbench coordinates (*use the Align button as required*).. ***The dimensions are corrected for any current asymmetrical scaling in View.***

Ctrl N or N Changes the arrowhead format of the selected dimension structure to automatically track the ***current Normal arrowhead format.*** All new annotation lines and dimension structures (*from boxes*) are set to automatically track the ***current normal arrowhead format***. The ***current normal arrowhead format*** is retained between program sessions.

Ctrl P or P Pastes the arrowhead format stored in the arrowhead format clipboard (*by Ctrl C*) to the currently selected dimension structure.

Ctrl R or R Reverses (*toggles*) the dimension arrow lines from inside (*one line*) to outside (*two lines*).

Ctrl U or U Uses the selected dimension structure's arrowhead format for the ***new current normal arrowhead format,*** and restores the selected dimension structure into tracking the ***current normal arrowhead format***. The ***current normal arrowhead format*** is used for all new annotation lines and dimension structures (*boxes*) and is saved between program sessions.



```
Box/Dimension Annotate Note:

Drag Edges to Size

Special Features Editing Keys:
  Any Arrow Key Lowers Note to Bottom of Display Chain
  Ctrl A or A  Toggles Arrow Points On and Off
  Ctrl B or B  Switches to Box From Dimension Mode
               Then Toggles Boundary Line On and Off
               If Off:  Blanking Corners Are Displayed
  Ctrl C or C  Save Dimension's Arrows as Clipboard Arrow
  Ctrl D or D  Switches to Dimension Mode
               (Point to Desired Box Edge FIRST Then Ctrl D)
  Ctrl F or F  Toggles Between Arrow Point Formats
               (Shift Left Mouse Button == Ctrl F)
  Ctrl L or L  Creates a Label Showing Selected Box Edge Length
               (Point to Desired Box Edge FIRST Then Ctrl L)
  Ctrl N or N  Change Dimension's Arrow to the Current Normal Arrow
  Ctrl P or P  Paste Clipboard Arrow as Arrows for Dimension
  Ctrl R or R  Reverses Dimension Arrow Lines Inside or Outside
  Ctrl U or U  Use Dimension's Arrows for Current Normal Arrow
               (Current Normal Arrow Saved Between Sessions)
```

Figure G-15 Help Screen for Annotate Label Objects

Editing Label Objects

Annotation labels have enhanced editing capabilities. ***For a summary help screen (Figure G-15), point to an annotation label and hit the <F1> key.***

Complex Labels (e.g. subscripts)

While the annotator doesn't support complex labels directly it can be easily tricked into making them. For example: Just insert a space where a subscript should go, create a label that is the subscript, and then drag it into the desired location.

Editing the Text of a Label Object

A label object is edited by pointing to it. A small rectangular cursor will surround one of the label's **letters** (*the action point*). Now edit the label as you would any normal text. Left and right arrow, insert, delete, and etc. keys work as in other editors.

Warning: Up or down arrow keys create a blank label. You will be asked if you're sure before the current text is blown away.

Creating Additional Label Lines

If you hit the **<Enter>** key, while pointing at an existing label object, a new label will be automatically created below the current label (*left justified with it*). This makes entering additional lines a snap. Remember! Each line is a ***separate*** label object. You need to move them as a ***group*** of objects (*discussed above*).

Optional Area Blanking

Normally a label is written directly over whatever is below. This can be a problem when there is clutter below. Thus it is often desirable to blank out (*erase with white*) the area below the label before writing the label. The blanking option provides this capability. To toggle this option, point to the desired annotation label and enter the **Ctrl B** key combination (*hold down the **Ctrl** key and press **B***).

Ctrl B Toggles annotation label area <u>B</u>lanking ***on*** an ***off***. Blanking is ***on*** when ***corners*** appear around annotation label.

Annotator Uses Windows Fonts with Windows Destination Devices

You can use **any** available Windows ***TrueType*** font when annotating ***if and <u>only</u> if*** one of the four Windows output destination devices is selected (e.g. *Windows Printer, Clipboard, Metafile, or Enhanced Metafile*). When a GUI printer language driver is active (*e.g. GUI PostScript*) ***<u>all</u>*** label objects will automatically revert to the GUI screen label size (*as per SIMION 6.0*). ***However, the Windows font selections are remembered and will be automatically restored when a Windows output destination device is <u>again</u> selected.***

Each Windows output destination options screen has a **Fonts** button. Click this button to access and select the font, type, and size you want to use as the ***current normal font*** for annotation labels (*remembered between program sessions*). All annotation labels are created to automatically track the ***current normal font***. When you change a font or font size existing annotation labels that track the ***current normal font*** will automatically be changed. The fonts appear on the screen as the size they will appear on output.

Individual Annotation Label Font Control

Each annotation label can have a different size and font when annotating when one of the four Windows output devices is selected (e.g. *Windows Printer, Clipboard, Metafile, or Enhanced Metafile*). Labels normally track the font and size of the ***current normal font*** described above. However, you have the option of using Ctrl keys to select and control fonts for individual annotation labels. The following is a list of the Ctrl keys and their functions. To perform a function point to the desired label and enter the desired Ctrl key.

Ctrl C <u>C</u>opies (*saves*) the font format of the selected label to the font format clipboard. This clipboard is local to the program. The **Ctrl P** function pastes this font format on selected labels.

Ctrl F Allows you to select a new <u>F</u>ont and/or size for the currently selected label via a Windows font selection screen. <u>Note: **Shift Left Mouse** button is same as **Ctrl F**.</u>

Ctrl N Restores the currently selected label into tracking the ***current <u>N</u>ormal font***.

Ctrl P <u>P</u>astes the font stored on the font format clipboard (*by **Ctrl C***) on the currently selected annotation label.

Ctrl S Shows the complete <u>S</u>ymbol table for annotation label's font. This is useful for selecting special characters. Just point to desired character and click a mouse button or hit the **Enter** key. <u>Note: **Shift Right Mouse** button is same as **Ctrl S.**</u>

Ctrl U Uses the selected annotation label's font for the new ***current normal font,*** and restores the selected label into tracking the ***current normal font. The current normal font*** is used for all new annotation labels and is saved between program sessions.

An Easy Way to Control all Arrow Formats

Annotation lines and dimension structures normally track the ***current normal arrowhead format*** automatically. The best (*and easiest*) way to change all arrow formats together is to point to an annotation line (*with an arrow*) or a dimension structure. Hit the **F** key until the desired arrow format appears. Now Hit the **U** key to change the ***current normal arrowhead format*** to the newly selected arrowhead format. Those lines and dimension structures that are tracking the current normal arrowhead format will automatically change to the new arrow format.

Restoring Annotations Via the RA Button (*if visible*)

If you have used any annotations in the previous printout. The GUI will provide an **RA** button (*Restore Annotations*) on the Annotate and Print Screen (*Figure G-12*) to allow you to restore the previous annotations (*very handy*). If you want to restore your previous annotations (*then rip out what you don't want*), be sure to click the **RA** button ***before*** creating any new annotations. ***If you don't, all the previous annotations will be erased!***

Annotations Can be Saved and Loaded as Files

Annotations can be saved and reloaded from files (*all fonts and formatting retained*). The files have an **.ANN** extension and the load and save buttons can be found on the Print Options Screen (*for an example see Figure G-10*).

Note: These buttons will ***not*** appear when the print function is called directly or indirectly from the file *manager (via a screen pixel print*). This prevents recursive calling of the file manager (*illegal*).

Exporting SIMION Graphics to Other Windows Programs

The addition of Windows support to the GUI has greatly facilitated the export of SIMION 7.0 output to other Windows programs. All the SIMION illustrations in this manual were created using the options described above. The figures were either saved to enhanced metafiles or transferred via the clipboard. The enhanced metafile route was most often used to reduce the rework time in case a figure was accidentally deleted from the document.

I have found that when creating illustrations in a program like Power Point, it is generally best to output the SIMION portion ***without*** annotations. The full power of the presentation program's illustrative features can then be used to maximize the balance and impact of the presentation graphics.

Editing SIMION output with another Windows program is possible (*since it is in metafile format*). However, ion trajectories and drawings of potential arrays can contain enormous numbers of line vectors. This can at times swamp (*dare I say sink*) the graphics editor. It is generally quicker to use SIMION to edit these features ***if possible*** than to dump them into an unsuspecting graphics editor.