



SwiftView

ICS Reference Manual

Revised for SwiftView 9.2.3.4

Table of Contents

<u>Chapter 1: The Imaging Command Set.....</u>	<u>3</u>
Manual Conventions.....	3
Product Specific Commands.....	4
Coordinate Systems	4
ICS parameter quoting.....	4
ICS Commands.....	5
<u>Chapter 2: SwiftView Environment Variables</u>	<u>44</u>
<u>Chapter 3: ICS Callbacks</u>	<u>47</u>
<u>Appendix A: PCL Fonts</u>	<u>53</u>
Overview	53
Font selection.....	53
<u>Appendix B: Paper size to PCL ID map</u>	<u>58</u>

Chapter 1: The Imaging Command Set

The SwiftView Imaging Command Set (ICS) is a simple set of English-like text commands controlling all viewing and printing functions. In response, many ICS commands generate "callbacks" which are easily parsed text strings. ICS commands and callbacks are accessed through UNIX pipes, Windows DDE or using other techniques discussed in the SwiftView Configuration Manual.

SwiftView's "GUI code" uses ICS commands and callbacks, so all internal viewer functions are available to outside programs.

Manual Conventions

All Boolean options accept any of the following: "on", "enable", "true", "yes", "y", "off", "disable", "false", "no" or "n", irrespective of the descriptions below. As of SwiftView version 7.0, "toggle" is also supported. Toggle switches the Boolean value to the opposite of its current setting.

In the ICS command summaries below, keywords are bold and data is not. Data may be numbers, e.g., **n** for an integer or **x1** for a real coordinate, or string values, denoted as **str** for a simple string, **fn** for a filename, or **ics** for a string of ICS commands.

Multiple options are listed within brackets [] and separated by a pipe |. In these cases neither the brackets nor the pipe should be included in the actual command, just one of the listed options. Example:

written in manual:	draw [all last]
ICS command:	draw all
	or
	draw last

Product Specific Commands

A number of the ICS commands are only available with a SwiftView Pro, or a SwiftConvert license.

Customers who purchased before SwiftView version 7.1.2 have what we refer to as a Traditional SwiftView license. Traditional SwiftView receives features that are no longer included in the Standard SwiftView; however do not receive all SwiftConvert and SwiftView Pro features.

Some commands are specified with a symbol in the brief description, these symbols denote that the command is only available in specific versions. The symbols correspond as follows:

P	SwiftView Pro
C	SwiftConvert
T	Traditional SwiftView
€	SwiftExtract

Commands that contain no special symbol are available in all versions: SwiftView, Traditional SwiftView, SwiftView Pro, and SwiftConvert.

Coordinate Systems

SwiftView commands and callbacks use three coordinate systems:

- 1) **Drawing coordinates**, where the units are the actual physical coordinates on the document as it would be normally printed. (When document files lack dimensions, SwiftView makes an assumption.) Values are real numbers and may have whole and fractional parts, e.g., 1.23. This is the default for markup.
- 2) **Pixel or "display" coordinates**, where the units are in pixels on the display. This is not used in markup. Values are integers only.
- 3) **Device coordinates**, where the units are the actual physical coordinates on the display or printed paper output. Values are real numbers. This is used only in markup.

In all cases, the coordinate 0, 0 is the upper left corner of the page and coordinates increase toward the lower right. Drawing and device coordinates are specified, and returned in callbacks, in inches or centimeters per the `units` and `markup units` commands. See the SwiftView Configuration Manual, Chapter 4, SwiftView Markup, for more details

ICS parameter quoting

Many ICS commands include strings which may contain spaces but which still must be single parameters. Unless otherwise noted, all such parameters must be "quoted" if they contain spaces in order to make them a single parameter. This is further complicated by the fact that ICS commands are enclosed by quote marks `"..."` on the command line or in other programming languages and ICS commands can contain other ICS commands as quoted string parameters.

For all cases except command lines, the sequence `\` should be used for all quotes embedded (nested) in a pair of `"..."`(quote marks). Here is an example:

```
gui button label "Hello" icon hello.bmp info "Demo tool tip" command "message \"Hello world\""
```

On a Windows or UNIX command line, an older method is to reset the ICS command quote character using the ICS command `"set quotechar ..."`, so that the outside quoting `"..."` pair is not confused. Here is an example:

```
sview -ics"set quotechar 0x5e | gui button label ^Hello^ icon hello.bmp info ^Demo tool
tip^ command ^message ^Hello world\^ ^"
```

The | (pipe) character can be used in all normal cases to concatenate multiple ICS commands.

A newline or dollar sign can be embedded in a command parameter string with the sequence "\^" or "\\$". This is particularly useful to make multiline messages.

In some cases, such as nested quotes, GUI commands, or ICS command strings that appear as a parameter of another command, the text is parsed multiple times and requires an extra backslash for a quote, or an extra carat or dollar sign for each extra parsing pass, e.g., "\\\", "\^^", or "\\$\$". Here are some examples:

embed a quote in a message in a configured ICS command:

```
set draw "message \"Check this \\\"draft\\\" carefully\"| draw all"
```

display a multi-line message:

```
message "line 1\\line 2"
```

or: `gui dialog message "line 1\\line 2"`

add a multi-line message hotspot:

```
markup circle rxloc 1 ryloc 2 radius 1 hotspot "message \\line 1\\line 2. \\""
```

select the next whitespace-terminated string using an ICS command sent via a pipe to UNIX SwiftView:

```
echo select string "\"(http| ftp)://[^\t\n\r]*\"" regexp yes > /tmp/pipe
```

(The pattern in brackets matches any number of space, tab, backslash, or newlines.)

ICS Commands

<ad>

Tick until last operation completes

Complete the previous command before processing the next one. HPGL file loading and all printing, saving and screen display do not complete for some time and can be interrupted by a subsequent command. If you want to submit another command without interrupting the last one, send this first to guarantee the earlier operation is complete. Any viewing command (e.g., `draw all` and `zoom in`) and any output command (e.g., `plot`) can also be interrupted.

browse

Send a URL to the browser

The URL is displayed in the last active window of the browser selected by the `set browser` command or in a new browser if none is running. Under the plug-in, this is the browser window that is displaying the plug-in instance, except that if two Navigator (not Communicator) windows are open, the wrong window may be used. The URL may not contain a #fragment field. "Browse" is commonly used with `gui button label` to add a button in SwiftView that will go to a website.

callback filename [fn| none]

Send callbacks to text file

Callbacks are generated as a result of various ICS commands, whether they come from the user interface (GUI) or from an external pipe. This command enables/disables appending those callbacks to a text file for use on Windows or UNIX. This can also provide valuable information for applications to use when incorporating with SwiftView.

dpage [n [first| last| next| previous]

Display page number n, first, last, next or previous

Load and display a new page of the current document. Equivalent to `lpage ... | draw`.

draw Draw default scale for current page

By default, this generates `draw widetop` for PCL files and `draw all` for all other file types. This default can be changed for all file types by using `set draw` described later in this manual.

draw [all| last] Draw all of the current page, or last requested view

`Draw all` clears the graphics display window, sets the scale factor so the entire current page will fit on the display, and displays the current page. `Draw last` is identical except that the current pan and zoom is retained.

draw [lasttop| lastbottom] Draw top or bottom of last page view

`Draw lasttop` and `draw lastbottom` preserve the current zoom and x position, but set the y position to the top/bottom of the page.

draw [top| bottom]

The default for these commands is `draw lasttop/draw lastbottom`. These commands are used when scrolling to another page in a `scroll up/scroll down` command. The default setting can be changed by using `set drawtop/set drawbottom`, described later in this manual.

draw [wide| widetop] Display top of page at full window width

`Draw wide` is a synonym for `draw widetop`.

draw widebottom Display bottom of page at full window width

get filename Return filename to callbacks

Report the current print filename in the `FILENAME` callback (set by the `set filename` command).

get plotters Return defined printers to callbacks

Report the list of defined printers in the `OUTPUT:CURRENT` callback.

get version Query SwiftView for its version.

Programmatically query SwiftView for its version and return to the callbacks.

gui autostart [enable| disable] Plug-in only, document viewed immediately

Equivalent to SwiftView plug-in HTML attribute `AUTOSTART="false|true"`. This is only relevant when using `gui inline disable`. If `gui autostart` is true then the SwiftView document will be viewed as soon as the HTML page is viewed. If `gui autostart` is disabled, a small icon will appear that the user will need to click to show the document. This will load the document in Standalone SwiftView, outside of the web browser. The default is `disable`. This command is ignored if `gui inline` is enabled.

gui borderwidth n Plug-in only, define solid border around plug-in

Equivalent to SwiftView plug-in HTML attribute `BORDERWIDTH=n`. This defines the width of the border around the entire plug-in. The default is 1 for embedded plug-ins and 0 for full-page. Note: if the available size allocated for the plug-in is 400 pixels in width and 300 pixels in height, a border of 50 pixels would cause the plug-in to be displayed in a 300 by 200 pixel area.

gui button label ls icon in info tts command ics

Create new button

Adds a button to be displayed to the right or underneath the previously defined controls. Each button control has a `label ls`, an optional icon whose file name is in `"(icon name)"`, an optional tool tip string `tts`, and an ICS command `command ics` to be executed.

If the current buttons are to be entirely replaced first precede this command with `"gui controls clear"`. An icon is specified by `icon in`, where `in` is one of the following icons built into the program: `exit`, `all`, `wide`, `rotate`, `reload`, `print`, `save`, `open`, `help`, or a filename pointing to a 16 color BMP file on Windows or an X pixmap file. Http URLs are accessible starting in SwiftView version 5.5.0. On UNIX `in` must be an absolute path, else relative to `$NDGUTIL` if it is set, else in the SwiftView program directory. The icon is centered and clipped to 20x20 pixels. If the icon is not specified and button icons are enabled with `gui controls style icons` the icon will be blank.

Example 1: Create a print button:

```
gui button label "Print" icon print info "Print document" command "gui dialog print"
```

gui clientmessage str

Send a message to the application

SwiftView does nothing with this command except return the rest of the command in a `SVC:GUI:clientmessage` callback. Client applications can use this to communicate through SwiftView, e.g., adding a command to the SwiftView user interface that is implemented in their own application.

gui controls clear

Delete existing buttons

Delete all existing tool bar buttons. The tool bar does not update until `gui controls update` is issued.

gui color colorname rgb:red/green/blue

Set Windows colors

Set a color for the user interface (Windows only). The default colors are the standard Windows colors, configurable by the user with the Windows display properties dialog except as noted. `rgb:red/green/blue` is an RGB triple (see markup attributes `bgcolor` for details).

Colorname is one of the following:

<code>background</code>	general background color (default Windows "3D Objects")
<code>button</code>	button 3D face (default Windows "3D Objects")
<code>shadow</code>	button 3D edge shadow (default as needed for Windows "scheme")
<code>buttonhighligh</code>	button 3D edge highlight (default whitened Windows "3D Objects")
<code>buttontext</code>	button text (default Windows "3D Objects" Font)
<code>buttonforeground</code>	button icon accent color (not settable from Windows, default blue)
<code>butttonhighlighttext</code>	selected button text (not settable from Windows, default blue)
<code>tipbackground</code>	tooltip background (default Windows "ToolTip")
<code>tiptext</code>	tooltip text (default Windows "ToolTip" Font)
<code>docframe</code>	frame around doc page (default Windows "3D Objects")

By default, except for `buttonforeground` and `butttonhighlighttext`, all colors depend on the user's Windows settings. The default Windows settings match the pre-SwiftView version 5.4 hard-wired colors, except button edge shading, which was black in pre-5.4 SwiftView. The scrollbar colors are derived from the button colors, except the arrows and dark 3D shadows are black.

Button icons loaded from a file are DIBs, and like the built-in button icons, the following colors are recolored per the above color settings:

<code>RGB(192, 192, 192)</code>	<code>button</code>
<code>RGB(0, 0, 128)</code>	<code>buttonforeground</code>

This recoloring may be inaccurate on displays configured with 256 colors or less, if the exact chosen button color is not in the current system palette. If you are trying to achieve a perfect color match, it is advisable to simply make the bmp the correct color in the first place, rather than making a gray icon and using SwiftView recoloring to match.

gui controls display [rightbuttons| topbuttons| none] Change location of control bar

Position the control bar at the right, top, or do not display it.

gui controls reset GUI controls and settings set to default.

Reset all controls and GUI settings to the default. Does not update the screen until `gui controls update` is issued. All icons defined by the GUI icon command are discarded.

gui controls style [labels| icons| both] Define button style

Configure the style of button controls: only text labels, only icons, or both. The standalone default is `both`. The plug-in default is `icons`. If none of the plug-in buttons have an icon, `both` reverts to `labels`.

gui controls update Update GUI

Update the controls according to all previously issued `gui controls`, `gui button`, `gui pagecontrol`, and `gui tooltips` commands. Controls are not updated when defined but wait until this command is issued to minimize screen movement. This command does not need to appear in ICS files, because it is issued at program startup, when all settings are complete, or after loading an ICS file. The only time this should be used is when SwiftView is being controlled via a DDE or pipe connection and the user interface is being changed after SwiftView has been started.

gui dialog activate Opens RLM new licensing activation window

For Windows only. This dialog can configure and install new licenses, as well as turn on roaming and set product configurations.

gui dialog installer Opens menu for changing file associations

This dialog allows the file type associations for Swiftview to be changed. Application must be run as Administrator.

gui dialog name Display a pre-defined dialog

<u>Dialog Name</u>	<u>Function</u>
copyright	Display version number, license status, and copyright information of the interface, which may be the ActiveX, plugin, or standalone.
copyrightsview	Display the version number of the underlying svview.exe (when the ActiveX or plugin is being used).
directprint	Copy the entire file directly to the printer. Allows selection of printer.
error	Display an error message.
fastprint	Print to a PCL or PostScript printer, bypassing Windows driver. Allows faster printing and mixed page sizes. (Same as print dialog on UNIX.)
fileprops	Display information about the current document and displayed page.

find	Search for a text string.																												
help	Display instructions for the basic SwiftView user interface.																												
info	Display information about the SwiftView product.																												
locale	Display the current system locale (language) setting. Not on UNIX.																												
message	Display an informative message. This command displays the string <code>str</code> in a SwiftView message box. User must click OK to close box. The message string can be multiple lines, separated by the sequence <code>\^</code> .																												
open	Select a file to view.																												
pagenumber	Jump to first, last, or specific page in document.																												
print	Universal print dialog, allows use of regular Windows printing, fast print or direct print according to the file viewed and the capabilities of the printer. In SwiftView versions prior to 5.5.0, "print" produces the winprint dialog.																												
P Save	<p>Save to file. Various output types supported.</p> <p>Using "def" before the option sets the default, allowing selection of other options. Otherwise the option is forced and others are unavailable.</p> <p>Options include:</p> <table> <tr> <td>[def]type <code>str</code></td> <td>-sets output type</td> </tr> <tr> <td colspan="2"><code>str</code> can be one of : original, pdf, tf_g4, text, tf3_192, tf3_96, or tf_unc</td> </tr> <tr> <td>[def]range <code>n-n</code></td> <td>-sets range to <code>n-n</code></td> </tr> <tr> <td>[def]onefile [true false]</td> <td>-sets 1 file or 1 file per page</td> </tr> <tr> <td>[def]scale [1 all printable]</td> <td>-sets scale factor</td> </tr> <tr> <td>[def]rotatedisplay [true false]</td> <td>-sets rotation</td> </tr> <tr> <td>[def]positional [true false]</td> <td>-sets vertical position</td> </tr> <tr> <td>[def]color [true false]</td> <td>-enables color or black and white</td> </tr> <tr> <td>[def]markup [true false]</td> <td>-passes markup or not</td> </tr> <tr> <td>[def]res [nnn original]</td> <td>-sets output resolution</td> </tr> <tr> <td>[def]filename <code>fn</code></td> <td>-allows output name specification</td> </tr> <tr> <td>showoptions [true false]</td> <td>-allows ability to show options button</td> </tr> </table> <p>Examples:</p> <table> <tr> <td><code>gui dialog save deftype PDF defrange all</code></td> <td>sets default type to PDF all pages</td> </tr> <tr> <td><code>gui dialog save type PDF range all</code></td> <td>forces type PDF all pages</td> </tr> </table>	[def]type <code>str</code>	-sets output type	<code>str</code> can be one of : original, pdf, tf_g4, text, tf3_192, tf3_96, or tf_unc		[def]range <code>n-n</code>	-sets range to <code>n-n</code>	[def]onefile [true false]	-sets 1 file or 1 file per page	[def]scale [1 all printable]	-sets scale factor	[def]rotatedisplay [true false]	-sets rotation	[def]positional [true false]	-sets vertical position	[def]color [true false]	-enables color or black and white	[def]markup [true false]	-passes markup or not	[def]res [nnn original]	-sets output resolution	[def]filename <code>fn</code>	-allows output name specification	showoptions [true false]	-allows ability to show options button	<code>gui dialog save deftype PDF defrange all</code>	sets default type to PDF all pages	<code>gui dialog save type PDF range all</code>	forces type PDF all pages
[def]type <code>str</code>	-sets output type																												
<code>str</code> can be one of : original, pdf, tf_g4, text, tf3_192, tf3_96, or tf_unc																													
[def]range <code>n-n</code>	-sets range to <code>n-n</code>																												
[def]onefile [true false]	-sets 1 file or 1 file per page																												
[def]scale [1 all printable]	-sets scale factor																												
[def]rotatedisplay [true false]	-sets rotation																												
[def]positional [true false]	-sets vertical position																												
[def]color [true false]	-enables color or black and white																												
[def]markup [true false]	-passes markup or not																												
[def]res [nnn original]	-sets output resolution																												
[def]filename <code>fn</code>	-allows output name specification																												
showoptions [true false]	-allows ability to show options button																												
<code>gui dialog save deftype PDF defrange all</code>	sets default type to PDF all pages																												
<code>gui dialog save type PDF range all</code>	forces type PDF all pages																												
toolbarprops	Change the toolbar and scrollbar properties.																												
winprint	Dialog for Windows print without Direct or Fast print options (version 5.5.0 and below).																												

gui dialog scrollmessage message str label str title str xpos n ypos n height n width n Scrollable message box

This command presents a more elaborate message box than the simple GUI dialog message command. It has a scrollable, selectable multiline message and a multiline label.

message

The `str` following is the text that appears in the scrolled window. Text will wrap automatically when extending beyond the window width. Use an escaped newline character to force a line break, or two escaped newlines to insert a blank line. To insert a newline, you must use the `\^` escape sequence required in `gui` commands. (`gui` commands are parsed twice, so two `^`'s are required if the command appears in a normal setting, three if the command is parsed multiple times e.g., in a markup command.)

label

The `str` following is the text that appears in a nonscrollable field above the scrolled window. Labels do not line-wrap; to make a multiline label, split the lines with an escaped newline character.

title

The `str` following is the dialog title bar text, which is clipped if too long.

xpos and ypos

The numbers following specify the initial location in pixels of the top left dialog corner, relative to the application main window. The default is 10,10.

height and width

The numbers following are the height and width in characters of the scrolled window. The dialog will be sized to accommodate this size plus the label text. If no size is specified, the default scrolled window size is a little less than half the width of the parent window, and approximately the same height as width.

gui dialog viewmode

Set the view mode for SwiftView

This command is only available when using the Windows version of SwiftView.

gui displayduplex [enable| disable]

Enables SwiftView to display duplex information

This command turns on the ability to display document duplex information. This information is displayed in the F8 dialog along with document properties, as well as adds a small binding mark on the edge of the paper if it contains duplex commands.

gui inline [enable| disable]Plug-in only, control if plug-in shows as part of HTML:
document

Equivalent to SwiftView plug-in HTML attribute `INLINE="false|true"`. If `gui inline` is enabled then the SwiftView document will be shown in the HTML document. If disabled, a **Show Document** button will appear on the HTML page and when clicked a separate "partner" SwiftView window will open and display the document. The default is `enable`.

gui key keyname ics

Define command for a key

gui key control keyname ics

Define command for a control-key or an alt-key

Keyname is one of the strings defined in the list below. The second column shows what each **keyname** represents as necessary. (For the benefit of X-Windows users, **keyname** strings are the same as X **keysym** names.) If "control" or "alt" is given, a control or alt key must be down when the key is pressed (control and alt are equivalent). **ics** can specify one or more ICS commands. Multiple commands must be separated by "|". Commands must be quoted if they contain more than one word. Some examples:

```
gui key Q          quit
gui key "Escape"  quit
gui key v          "plot 99 view"
gui key l          "lpage last| rotate 1| draw"
```

Upper and lower-case commands can have distinct commands, and case must be as shown for the keynames.

A key definition is removed by defining it to an empty command, for example `gui key A " "`. Note that the empty command must have a whitespace character, not just an empty quotation.

Keynames

A - Z		percent	%
a - z		ampersand	&
0 - 9		apostrophe	'
space		parenleft	(
exclam	!	parenright)
quotedbl	"	asterisk	*
numbersign	#	plus	+
dollar	\$	comma	,

minus	-	bracketleft	[
period	.	backslash	\
slash	/	bracketright]
colon	:	asciicircum	^
semicolon	;	underscore	_
less	<	grave	`
equal	=	braceleft	{
greater	>	bar or pipe	
question	?	braceright	}
at	@	asciitilde	~

"Home", "Left", "Up", "Right", "Down", "Page_Up", "Page_Down", "End", "F1" - "F12", "Print", "Insert", "Break", "Pause", "Scroll_Lock", "Sys_Req", "Escape", "Delete", "Return", "Num_Lock", "BackSpace", "Tab"

Keypad keys

"KP_Home", "KP_Left", "KP_Up", "KP_Right", "KP_Down", "KP_Page_Up", "KP_Page_Down", "KP_End", "KP_Insert", "KP_Delete", "KP_Multiply", "KP_Add", "KP_Subtract", "KP_Decimal", "KP_Divide", "KP_Enter", "KP_0" - "KP_9"

Additional keys not on PC keyboards:

"Begin", "Select", "Execute", "Insert", "Undo", "Redo", "Menu", "Find", "Cancel", "Help"

The following are not usable in Windows:

alt-anything, "Break", "Tab", "Print" (*Print-Scrn*), "Sys_Req", "control-Pause", "control-NumLock", "control-Scroll_Lock", "control-asciitilde", "control-Escape"

gui logo [enable| disable]

Show SwiftView bird icon in toolbar

Enable or disable the SwiftView bird logo icon, which appears between the regular buttons and the page control, and presents the copyright dialog when pressed. The default is enable for both standalone and plug-in. A -nologo option disables it in standalone.

gui logoicon [iconname| default]

Replace or reset SwiftView bird icon

Replace the SwiftView bird logo icon with the specified built-in button icon or file (see the **gui button** command for icon file requirements). Use **default** to restore the bird. The same icon is used for both horizontal and vertical control bars, and is clipped to 30 x 112 pixels and centered on the 30 x 112 button (new rules that also apply to button icon files).

gui mainwin ID

Determine if SwiftView has exited

If window ID goes away, SwiftView exits within 15 seconds. ID is generally set to the main window of the controlling application.

gui pagecontrol [enable| disable]

Turn off/on page display

This command can be used to disable the built-in "page N of M" display and the ability to jump to the first page, last page, or a specific page number. It always appears at the right/bottom of the toolbar. The default is enable.

gui scrollbars action [page| pan| both]

Define scrollbar display behavior

page For multipage files, dragging the scrollbar thumb changes pages while displaying a tool tip of the page number to be selected. For single page files, dragging the thumb (scrollbar) when zoomed in

- causes the view to pan within the page. Clicking the scrollbar shaft or end always pans, changing pages when appropriate. This is the default scrollbar action.
- pan** The scrollbar thumb performs panning only (when zoomed in).
- both** Same as the **default** (page mode), except that users can pan within each page of documents with less than 20 pages.

gui scrollbars display [vertical| horizontal| both| none] Define which scrollbars show

Enable display of desired scrollbars; the default is **both**. On Windows, the vertical scrollbar is always enabled unless disabled with this command. On Windows, the horizontal scrollbar appears only when zoomed in unless disabled with this command.

gui scrollbars dynamic [enable| pan| disable] Define when view will be updated

Indicates when view will be updated when dragging the vertical scrollbar thumb. The default is **pan**. This command interacts with **gui scrollbars action**.

Disable Never update the screen while dragging vertical scrollbar thumb.

Pan Update view of current page (new pan position) while dragging vertical scroll bar thumb but not when changing pages.

Enable Update view of page while either panning or paging with vertical scrollbar. A new page is displayed only after the thumb is still for 0.35 seconds.

gui tooltips [enable| disable] Control tooltips display

Enable or disable small information messages that are displayed when the user holds the mouse over a button. The default is **enable**.

gui toolbarprops [enable| disable] Control right click for properties option

Enable or **disable** the ability to use a right mouse click on toolbar. A right mouse click displays a pop-up menu allowing the user to change toolbar location and button styles. The default is **enabled** for the standalone. Issuing this command to the plug-in/ActiveX will enable the toolbar properties menu.

gui usercancel Cancel input mode with no action

gui userdef start Start user input mode

Similar to **gui userline**, but instead, SwiftView does no rubberband feedback while the mouse is down, merely changing the cursor. Instead, SVC:USERSHOW and SVC:USERDISPLAYSHOW callbacks are sent so the client can provide the feedback. The client draws the desired transient markup with DRAWMODE XOR, once on each callback SHOW or DISPLAYSHOW (but not both). The callbacks are sent twice for each position so that they will be erased by the XOR operation. SVC:USERDEF and SVC:USERDISPLAYDEF are sent with the final coordinates when the mouse is released. For best results client should XOR using a color that is the inverse of the final desired color, since it is **xoring** against a mostly white background. Also, use near-black for black, since black doesn't do anything in XOR on Windows.

gui userline start Start user input line mode

gui userpoint start Start user input point mode

gui userrectangle start Start user input rectangle mode

These commands cause SwiftView to display a "pencil" mouse cursor. Subsequently, holding down the left mouse button starts a "rubberband" of the specified entity. Releasing the left button terminates the

mode and generates the corresponding SVC:USER* and SVC:USERDISPLAY* callbacks with the user-defined drawing and display coordinates. Pressing the right button (or sending `gui usercancel`) causes the mode to be canceled with no further operation and generates the callback SVC:USERCANCEL.

gui winstate [hide| maximize| minimize| restore] Window Status

These commands enable the SwiftView window to be minimized and restored to the current user selection. `gui winstate hide` prevents either the window or the icon from being displayed (on UNIX, the window manager must implement `XwithDrawWindow()` handling). The command `gui winstate restore` returns the window to the previous displayed state, maximized or normal. `gui winstate maximize` is not implemented for UNIX.

gui wintitle str Change window title

This command changes the current window title to that in the string `str`. For example, you can display the name of the current document in the standalone SwiftView titlebar with the following command:

```
set ldocinit "gui wintitle \"\$docname()\" | rotate 0 | scale default"
```

hpgl [pen n| all] red ccc green ccc blue ccc width w Set HPGL pen color and width

N Pen number, 1-256
ccc Color component (RGB) setting, 0-255
w Pen width, defined as:
normal
bold
verybold
fine (always 1 pixel wide)
any real #, width in mm (e.g., 3.5)

hpgl startsoon [enable| disable] Enable early HPGL drawing

This command enables control of the initial display when loading an HPGL file. The default is `hpgl startsoon on` for standalone SwiftView, so lines are seen as they are read. When the drawing of the window is complete, if the initial scale was not correct, the view is then redrawn. This always leaves the drawing at `all`, reducing confusion about initial scale. The default is `Off` for the SwiftView Plug-in.

initdoc Run default ldocinit commands

Executes the commands defined by `set ldocinit`. This command is executed automatically at the end of an `ldoc`, and should be given after defining a document with "onpage use" commands.

ldoc fn Load a new document whose file name is fn

Load the document `fn`, but do not display it. The parameter `fn` can be a local file or an HTTP URL. Even standalone SwiftView can access files via HTTP in this manner including, through a proxy server. SwiftView automatically recognizes the file type. These include TIFF, HPGL, PCL, JPEG, CALS G4, PCX/DCX, ICS and SVP files, or zipped versions of any of these.

`ldoc` does the following:

```
send SVC:WORKING:ON callback
If set cachereset is enabled, resetcache
```

If `fn` is an ICS file:

```
send SVC:WORKING:OFF callback
parse the file
```

If `fn` is not an ICS file:

- use `reset` except for `resetcache` and setting current document name
- set `page count`, or initiate background page counting for file types that require it
- set the current document name to the outermost enclosing ICS filename, or the `fn` if the `ldoc` command is not in an ICS file
- clear all `onpage markup` commands and transient markups
- `lpage 1`
- `initdoc`

`lpage [n| first| last| next| previous]`

Load page

These commands load but do not display a page of the current document. A `draw` would be the typical next command (see `dpage`). This enables the IS professional to use `ldoc`, `lpage`, `draw` to display page 2 or higher in a document first. `lpage` does the following:

- send `SVC:WORKING:ON` callback
- load the specified page
- clear transient markups
- send `SVC:WORKING:OFF` callback

`markup attributes coordinates c xorigin n yorigin n`

Establish markup coordinate system

<u>Parameter</u>	<u>Legal Data Description</u>
<code>coordinates drawing</code>	Use drawing coordinates
<code>coordinates device</code>	Use device (paper / screen) coordinates
<code>xorigin + nto -n</code>	Move the coordinate origin relative to the left (+ values) or right (- values) edge.
<code>yorigin + nto -n</code>	Move the coordinate origin relative to the top (+ value) or bottom (- values) edge.

`markup attributes defaults`

Set default attributes

This resets all parameters to the same initial values loaded with each new page. These are equivalent to the following:

- `display yes`
- `print yes`
- `coordinates drawing`
- `xorigin 0`
- `yorigin 0`
- `fgcolor defaultforeground`
- `bgcolor defaultbackground`
- `foreground defaultforeground`
- `background defaultbackground`
- `drawmode opaque`
- `linestyle normal`

`markup attributes display [yes| no]`

Display or print only

Defines whether markups are to be displayed on the screen. The default is "yes", while "no" causes all subsequent markup commands to be added as print only (no display).

`markup attributes drawmode xor`

Draw in xor - for dynamic feedback when editing markup

The foreground and background colors in the markup are **exclusive-ord** with the original. Resulting colors may be unpredictable, but it is guaranteed that if the same markup is drawn twice, the original will be restored. Note: this command is for transient markups only.

markup attributes linestyle ls fillpattern fp fillshade fs

Display or print only

<u>Parameter</u>	<u>Legal Data Description</u>
linestyle normal	Normal lines
linestyle bold1	Bold lines
linestyle bold2	Very bold lines
fillpattern solid	Solid fill
fillpattern shade	Shade set with "fillshade"
fillpattern densedots	Other fill patterns
fillpattern mediumdots	
fillpattern lightdots	
fillpattern boxes	
fillpatter weave	
fillshade 0-100	where 0 = no shade, 100 = continuous current color

NOTE markup attributes linestyle does not apply to text markup.

markup attributes replaceonly colorname

Replace only a given color

Draw only where the existing color is colorname. Currently, only colorname pixels in the drawing are guaranteed to end up as the markup color, and only if colorname is black or white. See markup attributes fgcolor for a description of colorname parameters.

markup attributes dontreplace colorname

Don't replace a given color

Do not draw where the existing color is colorname. Currently, only drawing pixel values that are the opposite of colorname are guaranteed to end up as the markup color, and only if colorname is black or white. See the example below for using this attribute to highlight non-background pixels in the document. .

markup attributes dontdraw colorname

Don't draw in a given color

Do not draw if markup color is colorname. Currently, only source colors that are colorname are guaranteed not to be drawn, and only if colorname is black or white. See markup attributes drawmode transparent below for using this attribute to add transparent markup images.

markup attributes drawonly colorname

Draw only if using a given color

Draw only if markup color is colorname. Currently, only source colors that are colorname are guaranteed to be drawn, and only if colorname is black or white.

markup attributes drawmode opaque

Set markup to opaque

The foreground and background colors in the markup replace and cover up the original document image. This is the default markup rendering attribute.

markup attributes drawmode transparent

Set markup to transparent

This is the rendering mode normally used to fill an area with a color while allowing the existing image to remain visible, e.g., to create "transparent sticky-notes". It is equivalent to markup attributes replaceonly defaultbackground. Note that the color that results when the existing image is a color other than black or white is undefined, but on most systems will be a color "between" the existing and markup colors.

markup attributes bgcolor colorname Set background markup color

markup attributes fgcolor colorname Set foreground markup color

These commands set the colors used for all markup foreground and background drawing, respectively. **bgcolor** is used for the background of text (except the "ascii" face) and markup load of bi-tonal images. **fgcolor** is used for all other markups. **colorname** in all color attributes is one of these:

defaultbackground When setting **fgcolor** use the current **bgcolor**, otherwise use white (rgb:ff/ff/ff).
defaultforeground When setting **fgcolor** and **bgcolor** is black, use white, otherwise use black
background The color defined by "markup attributes background . . ."
foreground The color defined by "markup attributes foreground ..."
rgb:< red> /< green> /< blue> An RGB triple, where < red> ,< green> and < blue> are the 8-bit hexadecimal values 00-ff.

The default colors are **defaultforeground** (black) for **fgcolor** and **defaultbackground** (white) for **bgcolor**. NOTE: In order to ensure reasonable rendering on bi-tonal devices (TIFF, CALS, PCX, HPLJ and HPGL-2 printers), any colored background fill (such as for a "sticky note") would be drawn with **bgcolor= fgcolor**. This ensures that the area is cleared to the background color (white) on bi-tonal devices.

markup attributes background colorname Define colorname "background"

markup attributes foreground colorname Define colorname "foreground"

Define the values of the color names **foreground** and **background** used in markup attributes **fgcolor**, etc. The default values are **defaultforeground** and **defaultbackground**. This command only affects subsequent uses of the color names **foreground** and **background**.

Declaring **foreground** and **background** as the normal foreground and background colors of a particular document ensures that built-in color operations will function properly. For example, if the drawing's background color is black, setting **markup attributes background rgb:0/0/0** (black) will ensure that the automarkup-generated "URL hotlinks" work correctly. (Currently this only works for black or white backgrounds).

markup attributes print [yes| no] Enable/disable printing of markups

Default is yes. This command allows disabling the printing or saving of all markups.

markup attributes background push Push attributes

markup attributes foreground pop Pop attributes

Push/pop the current markup attributes to/from a stack. This current markup CDP (position), pagesize, markup load scale, and font selection are not affected.

markup entity hotspot ics status str Add hotspot to graphical entity

The optional **hotspot** parameter defines a set of ICS commands to be issued when the user clicks on the markup entity. When the cursor is over a hotspot region it changes to a "pointing index finger".

If the **status** option is given, when the cursor is over the hotspot, the status "string" is displayed in the browser status line in the plug-in or in a tooltip window that appears at the lower left corner of the standalone SwiftView window. If the status **str** is empty, the hotspot ICS commands are displayed. If the **status** option is not used, no text status display is performed.

The markup attributes display/print commands affect whether the geometric entity is visible, but do not affect dynamic user feedback or the hotspot action. Usually you will want to set markup attributes display no "print no" when adding a hotspot to an existing drawing region.

hotspot may be applied to any of these markup entities: rectangle, filledrectangle, polygon, filledpolygon, circle, filledcircle, text.

markup circle rxloc x ryloc y radius r Draw circle

markup filledcircle rxloc x ryloc y radius r Draw filled circle

Draw circles and filled circles. Filled circles have only the fill, no edge. The parameters rxloc and ryloc are optional, but if used change the current drawing position (and use the CDP if omitted). Radius is required. NOTE: hotspot, documented above, is an optional parameter.

markup clear transient Clear all transient markups

Deletes any transient markups, but not any onpage markups (does a markup clear all, then reloads the page including onpage markups.)

markup clear all Deletes all visible markups

Deletes all visible markups on the current page, whether they are transient or onpage. Note that onpage markups will be redrawn if the page is reloaded. Use of markup clear transient, onpage reset, or markup reset is preferred to this command.

markup clear last Clear last markup

Clears the last markup applied from either a transient or onpage markup command, and is useful for markup editing.

markup crop left lmargin right rmargin top tmargin bottom bmargin Crop from edge of page

markup crop rrect x0 y0 x1 y1 Crop using a rectangle

Any combination of left, right, top and bottom parameters may be specified to clear (crop) the specified edge region of the current page. The margins are the widths of the areas cropped. rrect defines a rectangle outside of which everything is cropped. This is provided as a convenient way of eliminating extraneous information from the user's view. NOTE: These commands do not affect CDP or markup attributes.

markup display markups Display only markups

markup display original Display only original

markup display both Display both markups and original

These commands determine how the current page is displayed by the draw commands, how the current page is printed with the print command, and how the current page is saved with the save command. The default is display both, which shows both the original document and its markups. Using display original shows only the original document content, while display markups shows only the markups. Note that these commands apply globally to all markups, unlike markup attributes display| print, but, like all transient markups, they are cleared when a new page is loaded.

markup line x0 y0 ... xn yn Draw line

markup arrow x0 y0 ... xn yn Draw arrow

Draw a multi-segment line one pixel wide or line with arrowhead as defined by the set of (x, y) coordinate pairs. The coordinates xn yn become the CDP, as in this example:

```
markup line 1 1 5 3
```

```
markup load filename fn rxloc x ryloc y scale s rotate r Load file as markup
```

markup load filename fn rxloc x ryloc y scale s rotate r Load file as markup

This command should always be used with the “onpage” command to specify the page to load on. All parameters are optional except "filename", a single page TIFF, JPEG or CALS G4 type 1 file. If a destination location is not supplied, the CDP is used, and, if a destination location is provided, it changes the CDP. If a destination scale is not provided, the last value provided is used (1.0 is default). This scales the incoming image by the amount specified (0.5 = 1/2 the size). The incoming file's resolution with respect to the original file's resolution affects the displayed and printed size of the markup image. The position of the image's upper left hand corner is specified by rxloc and ryloc.

markup pageinit Set all markup modals to defaults

This command is implicitly executed by SwiftView whenever a new page is loaded, but can also be explicitly executed at any time. It does the following:

```
Sets the CDP to 0, 0
```

```
markup attributes defaults
```

```
markup text defaults
```

markup polygon x0 y0 ... xn yn Draw polygon

markup filledpolygon x0 y0 ... xn yn Draw filled polygon

Draw polygons and filled polygons. Edge fill works the same as rectangles. Coordinates xn yn become the CDP. NOTE: "hotspot", documented above, is an optional parameter.

markup rectangle x0 y0 x1 y1 Draw rectangle

markup filledrectangle x0 y0 x1 y1 Draw filled rectangle

Draw rectangles and filled rectangles. Filled rectangles have only the fill area (i.e., no edge rectangle). In order to get both an edge rectangle and the fill pattern, two commands must be used. Coordinates x1 y1 become the CDP. NOTE: hotspot, documented above, is an optional parameter.

markup reset Resets markups

Clear and discard all transient and onpage markup commands, then redraw the page.

markup text font "... " rxloc x ryloc y string "... " Draw text

This command draws text in the markup. All parameters/data pairs are optional and may be in any order, but at least one pair is required. Text is written only when the string parameter/data pair is provided. Except for hotspots, data not provided with string are taken from the last setting. The coordinates rxloc ryloc, the x and y locations, become the CDP. If rxloc, ryloc are omitted, the CDP is used in their place.

string is the actual text and must be enclosed in quotemarks ("..."). The "string" becomes a "text entity" of the designated font in memory. The actual font displayed may be the internal stroke font (ASCII) or Courier fonts obtained from files provided with SwiftView. The actual font printed is provided and selected by the printer itself.

font sets the current font and its data must be a single parameter in quotemarks ("..."). The font data string defines the current font with a set of "command / data" pairs separated by spaces. The default setting is equivalent to `face ascii style normal size 120 pitchcpi 10`. Any combination of font commands may be specified in any order and only one is necessary to change that single aspect of the font. The font commands and legal data are:

Font Command	Legal Data	Description
face	cour	Typewriter font
	ascii	SwiftView stroke "engineering drawing" font
	sans	Sans Serif proportional font
	serif	Serif proportional font
style	normal	Normal
	bold	Bold
	italic	Italic (no function for face ascii)
size	1. 0 - 100.0	Height of font in points
pitchcpi	1 - 40	Fixed pitch in characters per inch

The text defined by the string parameter can include `$strftime()` containing parameters used by the standard C library `strftime()` call. The results of that call, as defined in system documentation, are substituted in the string when the command is loaded. Here are some examples of parameters:

```
%c   Date and time representation appropriate for locale
%x   Date representation for current locale
%X   Time representation for current locale
```

EXAMPLES:

```
markup text rxloc 2 ryloc 3           Next string will be at x= 2, y= 3
markup text font "style bold size 24" Next string will be bold 24 point
markup text string "This is the string" Writes the string with current values
```

NOTE: You can also use them all at once, as below. Note also that `hotspot`, documented above, is an optional parameter.

```
markup text rxloc 2 ryloc 3 font "style bold size 24" string "This is the string"
```

markup text rotate n Rotate orientation of markup

Rotates orientation of markup with respect to base page. Allowable values of `n` are 0, 1, 2, and 3 corresponding to 0, 90, 180 and 270 degrees.

markup units [inches| centimeters] Set markup units to inches or centimeters

This defines how dimensions are treated in all subsequent SwiftView markup commands. Changing units has no effect on previously executed commands. Default is `inches`. This command applies only to markups – see the units command to set the units of other ICS commands.

onpage ppp ics Apply markups to defined pages

Onpage ICS commands are processed after the document page is completely drawn, and are processed in the order given. The order of processing of "transient" markups and onpage commands is not specified.

`ppp` defines which pages the ICS command applies to. Legal values, include `all`, any single number, ranges of numbers of the form `6-9` and comma-separated lists of single pages and ranges like `1,5,7-11`.

ics defines the ICS commands to be applied. Any ics commands may be given.

onpage ppp use sourcename fn lpage [enable| disable]

onpage ppp use blankpage size lpage [enable| disable] Define pages of the document

onpage [odd| even| front| back| range 1-n| simplex| duplex| long| short] Customizable onpage commands for duplex.

onpage duplex [front| back| long| short]

onpage duplex long [front| back]

onpage duplex short [front| back]

Duplex and Simplex options only apply to PCL files.

Onpage use commands define the given pages of the base document, including the page size and its basic (pre-markup) contents. Onpage commands may be issued at any time, incrementally changing the current document. Defining page 1 with **onpage use** automatically does an **lpage 1** command, unless **lpage disable** is given. (The **lpage** option is new in versions 5.2.1 and higher.)

sourcename fn defines the page to be the first page of the given document file. Any supported type of file may be used (see **ldoc**), and pages may be of mixed types. The **fn** argument may contain one integer of the form **%[n]d**, which is replaced by the current page number.

blankpage size defines the size of the specified pages, and clears any base document data on the pages. **size** may be one of the following: **letter**, **legal**, **B**, **C**, **D**, **A0**, **A1**, **A2**, **A3**, or **A4**.

NOTE: Direct Printing is not supported by virtual documents created with **onpage use** commands.

onpage reset Reset all onpage commands

Use this command to "unload" the current document. It clears all **onpage use** and **onpage markup** commands and transient markups, and clears all states associated with an **ldoc**. Equivalent to **use reset** plus clearing all markups.

pagesize rxsize xs rysize ys Set new page size

pagesize original Restore original page size

This defines the size of the current page to the number (**xs ys**) or to the size of the "original page" (i.e., the first image or drawing loaded) for markup commands. This must match the current page's size in both dimensions for all markup commands to work correctly. This value is reset to the value defined by the input file each time a page is loaded. This command is used for markups only and has no impact on the size of pages that are generated by **plot** or **save** commands.

pan [up| down| left| right] Pan view accordingly on drawing by 25% of display

pan [x p] y p Pan in x or y by real number p

pan x p and **pan y p** control panning in each dimension defined by a real number **p** (0.0 - 1.0).

pan x 0 means put the left edge of the drawing at the left edge of the display area.

pan x 1 means put the right edge of the drawing at the right edge of the display area.

pan y 0 means put the bottom edge of the drawing at the bottom edge of the display area.

pan y 1 means put the top edge of the drawing at the top edge of the display area.

pcl defaultres [300| 600] Set default display resolution

P 

This command specifies the default display resolution for PCL5 documents that do not contain resolution commands. The default setting for SwiftView is "pcl defaultres 300"

pcl defaultduplex [long| short| original] Set default duplexing behavior if duplex is enabled

This command specifies the default duplex behavior when "pcl duplex" is enabled. This is similar to setting the front panel of a duplexing printer. Original specifies to not force duplexing, but follow the commands in the file.

pcl duplex [enable| disable] Determines if SwiftView emulates a duplex or simplex printer

"pcl duplex on" tells SwiftView to emulate a duplex printer. "pcl duplex disable" emulates a simplex printer, ignoring duplexing commands in the document and "gui displayduplex on" has no effect. The default is "pcl duplex on".

pcl edgetoedge [on| off] View edge-to-edge

This command changes SwiftView operation so that PCL files with text to the edge of the paper will view and print as edge-to-edge on printers that support this feature. Only documents loaded after the command is given are affected. If viewing an edge-to-edge document and printing on a printer that does not have edge-to-edge support, text inside the non-printable area will be cut off.

pcl forceres [0| 300| 600| 1200] Force default display resolution

P 

Forces the default display resolution for PCL5 documents, regardless of whether they contain resolution commands or not. "pcl forceres 0" restores the pcl defaultres setting.

pcl initfile fn Expands PJP & PCL configuration

Allows configuring SwiftView's default PCL settings beyond what is possible with the documented PJP DEFAULT commands and the PCL defaults dialog. SwiftView will now read a PCL5 initialization file which is processed after each explicit or implicit PCL5 ESCAPE-E reset command contained in a PCL 5 document as well as at the beginning of any PCL file.

pcl linefeed [cr_lf| lf] Affects how SwiftView treats linefeeds/carriage returns

PCL printers allow configuring the meaning of the carriage return, linefeed, and formfeed commands, so that carriage return implies a linefeed, linefeed and formfeed imply a carriage return, or both. SwiftView defaults to adding carriage returns (cr_lf) if it detects that the file is a unix text file, else defaults to the PCL printer default of assuming no extra implied behavior. This command overrides the setting based on file type: "cr_lf" adds a carriage return; "lf" adds no extra implied behavior.

pcl [letter| legal| a4| a3]_out_tray [n] default] Specify tray # to use for certain paper sizes

These commands only affect Fast Print, and are disabled by `pcl passpapersize off`. Fast print can add tray selection PCL commands for printers that cannot handle the PCL commands that specify the paper by size (i.e., Letter, Legal, etc.) and instead must be specified by tray number (e.g., Tray number 5). Printers known for this problem include the HP 2100, Brother HL-P2500, and Lexmark E320. Of course, this only helps if there is a known correspondence between trays and paper size.

The default value, restored by `pcl [letter|legal|a5|a4|a3]_out_tray default`, is set by the universal print dialog to the pre-SwiftView version 5.6 default settings:

```
letter/A4      5
legal/A3      1
```

for printer drivers named `"*HP*2100*"`, `"Brother P2500 series"`, or `"*HP LaserJet IIP*"`, in "Use document paper sizes" mode.

For all other printers and at program startup, the default is set to 0, which disables sending tray commands. No tray selection is added except for these four paper sizes. For example, to associate the upper tray with letter and the lower with A4:

```
pcl letter_out_tray 1
pcl a4_out_tray 4
```

NOTE: if a file has tray selection commands and `pcl passtray` is enabled, the tray commands from the file will be output first, then the tray selection generated from the paper size, then the paper size command. Because on most printers the last command wins, most printers that can respond to paper size will ignore all tray settings, and tray selections generated from the paper size will override tray settings from the document.

pcl oldfonts off Better match modern printer boldness

Fixed-width fonts of several sizes are displayed too large and bold for modern printers. This command enables SwiftView to match boldness to newer printers by disabling boldfacing for Windows line printer scalable fonts and deferring SwiftView's compiled-in bitmap fonts when a close True Type is found. This feature will not work on UNIX until SwiftView is delivered with scalable fonts.

pcl pagesize n orientation o [linesperpage| fontpitch] p reset Define page specifications

This sets the default page size for viewing PCL files and is equivalent to front panel settings on an HP LaserJet printer, where parameters are defined as follows.

n =page size codes:

1	Executive	911	Pressure1up (3x6.5")
2	Letter	912	Pressure4up (12x14.9")
3	Legal	913	Sackbag (7.5x14.9")
6	B size (Ledger)	920	(17x12.5")
25	A5	921	(13.6x8.4")
26	A4	922	(17x4.3")
27	A3		
910	Cheshire (11x14.9")		

o = page orientation: landscape or portrait. The default is portrait.

l = lines per page: must be a positive integer. The default is 60.

p = pitch of fixed pitch fonts must be a positive real number. The default is 10.

reset resets all `pcl` command settings to the factory default front panel setting. This is equivalent to sending `pcl pagesize 2 linesperpage 60 orientation portrait fontpitch 10`

pcl passduplex [enable disable]	Pass duplex commands to printer
pcl passtray [enable disable]	Pass tray commands to the printer
pcl passpapersize [enable disable]	Pass paper size commands to the printer

These commands control pass-through of duplexing, tray and paper size selections from the original PCL file. The default settings are 'on' for duplex and papersize, and 'off' for passtray. If printing using the Universal Print Menu, papersize can be controlled using the Print Method, Properties, and Options.

By default (**pcl passduplex enable**), duplex commands present in the original file are passed to the printer and printer driver duplex settings are ignored. To use the duplex settings in the printer driver properties, you must send SwiftView the command **pcl passduplex disable** before loading the document. These commands have no effect if Direct Print is the selected Print Method.

PCL **passpapersize** affects HPLJ3_ORIGINAL fast printing and Windows printing, but has no effect with the HPLJ3_<size> drivers. However, not all Windows drivers respond to paper size commands correctly, so **passpapersize** may not work. With **passpapersize** off, Windows printing obeys the driver settings; HPLJ3_ORIGINAL defaults to requesting letter (unless overridden by **passtray**). The fastprint dialog uses only HPLJ3_ORIGINAL; the universal print dialog uses HPLJ3<size> in "Use driver size property" mode.

plot pn [all m-n current rangefile:fname] [cut x1 y1 x2 y2] rcut x1 y1 x2 y2 view odd even reverse] [onefile]	Plot specified output to printer
---	----------------------------------

€

This command prints (plots) as defined for the printer number using "printer ...". If the "printer command ..." defined is "FILE", then the printing is done to the file name specified by "set filename". In all other cases, a temporary file is written on UNIX for each page and the "printer command" applied to it as described elsewhere in this manual. Note that "printer types" include TIFF files, enabling printing of TIFF files. Parameters are:

- pn Printer number: 0-89. Printers 90-99 are reserved for use by SwiftView. Print to default Windows printer by using 99.
- m-n Print pages m through n inclusive
- x1,y1 Upper left of coordinate rectangle
- x2,y2: Lower right of coordinate rectangle
- cut and rcut define a rectangle in pixel or drawing coordinates (real in/cm per the units command).
- rangefile:fname specifies to load a text file with comma delimited page ranges. This is used for specifying large complicated page ranges.
- onefile used only when outputting to a file. Specifies to create one file and not one file per page

The plot command can also be used to specify input trays. Do this by finishing a page range with tX where X is the tray number. The following example grabs pages 1-5 from tray 1, and pages 6-10 from tray 2:

```
plot 12 1-5t1,6-10t2
```

It is important to note that tray numbering is specific to printer models. Many printers refer to their output trays differently, so the tray specifications may need to be adjusted for each printer.

printer number pn type pt command cs alias as	Define printer type and how to print.
--	---------------------------------------

**Print output types have changed with 8.1.0 . The previous notation for output types will still be recognized, however they are no longer documented in this manual. If you use a version of SwiftView older than 8.1.0 please contact tech@swiftview.com for the older manual.*

This command allows you to define an output printer for printing. This allows automated printing from the command line, or setting up printing in UNIX environments. These commands specify the printer number, output type, any special commands to be passed, and the "alias" for pointing to printers using their name or port number. Paper output size, duplexing, resolution, and color output settings are set using the `set printsize`, `set printduplex`, `set printres`, and `set printcolor` commands listed later in this manual.

pn Printer number (0-99).
pt Printer output type from the following list:
DIRECT Copy the original file directly to the printer.
PCL5 SwiftView PCL5 RASTER output. Default output type for FAST print
POST2 PostScript Level 2 using Group 4 PostScript filter
RHPGL_* HPGL-2 plotters with HP's RTL
 * can be any of the following sizes: A, B, C, D, E, A0, A1, A2, A3, A4
 Example: RHPGL_A
 The default model is HP250. Other HP plotter types are set with the environment variable NDGHPGL.
 See "Optional Environment Variables".
RHPGL_US Pick closest of A-E to the original drawing size.
RHPGL_EURO Pick closest of A0-A4 to the original drawing size.
CALS CALS Raster Type 1 file
MS_BEST Duplicate the printing options specified in the print dialog when the print method has been "reset to default".
MS_WIN MS Windows printer driver identified by alias.

cs UNIX: System command string printing the temp file
 UNIX and WINDOWS: `FILE` causes print to default file name (use the command `set filename ...`) "none" should be the default otherwise.

as Alias string displayed for user on UNIX, Windows share name of printer
 On MS Windows this is used to contain 3 component strings defining the printer to be used. This alias can also be used as the network path of a shared printer, such as `//computername/printername` .

printer default pn Define default printer number

This is the printer displayed as the default on the X Window output menu and used by the `plot` command when no other parameters supplied. It has no effect on the Windows GUI plot menu and, in turn, is not affected by that menu. `pn` is the Printer number (0-99).

quit SwiftView program exits

reload Reload document

The `reload` command `re-ldocs` the current document and redisplay the current view of the current page. The current document is the most recent `set docname` command, or if empty or not given, the most recent of:

- 1) a non-ICS file loaded outside an ICS file
- 2) the outermost ICS file that (directly or by loading another ICS file) either loads a non-ICS file or contains `onpage reset` or `use reset`
- 3) an `onpage/use reset` not in an ICS file = no current document. The only `ldoc` action performed is to issue `resetcache` if the `set cachereset` option is `readonly`.

In Windows SwiftView controls (e.g., plug-in), `reload` re-downloads all files from the web server, reinitializes the user interface, and sends a `reload` command to SwiftView. This is not implemented in the UNIX plug-in. By default, `reload` is assigned to the F5 and ^R keys. A `reload` button icon is included so you can add a `reload` button if desired.

resetcache Close all cached files

This is done automatically on an ldoc command unless set cachereset disable is given. The file data is not discarded from the cache.

resetpages Recalculate the number of pages

This command recalculates the number of pages in the currently loaded document and updates the main window display.

rlminstall [product {productname}] file {file/url} type {type_num} Installs RLM licensing information

type_num=0 | rlm rlm.lic -- for client licenses
 type_num=1 | host host.lic – for host/client licenses
 type_num=2 | demo demo.lic – for custom demo license installs
 type_num=3 | roam roam.lic – for roaming license entries

This command is used to install SwiftView with the RLM based licensing model.

EXAMPLE: rlminstall product swiftview_pro file 12345.lic type rlm
 rlminstall product swiftview file customer.lic type 0

rotate [n] next] Set rotation angle for all subsequent viewing commands

To complete a rotate requires a draw to redraw the current page at the new rotation. Rotate is for display only, and does not affect printing. The parameter n is defined as:

0 = normal (what's saved in the file)
 1 = 90 degrees
 2 = 180 degrees
 3 = 270 degrees
 next = sets the rotation to the next clockwise value.

save ft [all| m-n| current| rangefile:fname] fn [dcut x1 y1 x2 y2| rcut x1 y1 x2 y2| view| odd| even| reverse| onefile] Save

Ⓞ € Ⓣ

This command is used to save the loaded PCL file in some other format. It allow you to save the whole document, a page range, or a cut area.

Paper output size, duplex, resolution, and color output settings are set using the set printsize, set printduplex, set printres, and set printcolor commands listed later in this manual. In addition, PDF, TIFF, and PNG allow different compression rates. The compression is set using set printcompressformat also listed later in this manual.

NOTE: HPGL file type values can be found under the “printer number” command

Legal values of file type ft are as follows.

€	ORIGINAL	Original output type, used for direct extraction of PCL page ranges
Ⓞ Ⓣ	PCL5	RASTER PCL5 output
Ⓞ Ⓣ	TEXT	ASCII text (onefile, n, rcut, cut, view options are usable but not 'all').

		Form feeds in the PCL source file will be saved in the file.
Ⓞ € ¶	TEXTPOS	See below.
Ⓞ € ¶	TIFF	TIFF
Ⓞ € ¶	CALS	CALS Raster Type 1 file (the all1 option is not supported for CALS)
Ⓞ € ¶	PCX	PCX file. "onefile" writes a multipage PCX file (DCX file).
Ⓞ	PDF	PDF
Ⓞ	PNG	Monochrome PNG output
all		indicates all pages of the document.
m-n		is a page range; it . can be specified, forward, backward, and separated by commas, for example, 5-7, 7-5, 5,9,3
current		indicates only the current page.
rangefile:fname		specifies to load a text file with comma delimited page ranges. This is used for specifying large complicated page ranges.
odd, even and reverse		specify all the odd pages, all the even pages, and printing in reverse order.
dcut and rcut		define a rectangle in pixel or drawing coordinates (real in/cm per the units command) which is saved to the defined file name and type. x1, y1: Upper left of coordinate rectangle. x2, y2: Lower right of coordinate rectangle.
fn		specifies the output filename.
onefile		specifies to save the document as one file, instead of individual files per page.

By default, when saving multiple pages, SwiftView saves each page individually naming it filename.001, filename.002, etc. . . If you add the optional command **onefile** to the end this specifies to save the entire range to 1 file.

Examples assuming a 3 page document is already loaded:

save TIFF all doc1	saves one TIFF file per page, doc1.001, doc1.002, and doc1.003.
save TIFF all doc1 onefile	saves one multi-page TIFF file, doc1.
save TIFF 2-3 doc1	saves pages 2 and 3 as doc1.002 and doc1.003.

save textpos option fn

Save text position information

Ⓞ € ¶

This follows the same format as the save command above and has the same options. Saves the position of each word to a formatted file for external analysis. It writes one line for each page from which text is saved:

page n

followed by one line for each selected text item:

rx1 ry1 rx2 ry2 string

where the four coordinates are the rendered bounding box of the given text string, in the current real drawing units (inches/cm from upper-left corner of document per the units command), to a resolution of 0.001 inches. (The coordinates and the strings are the same as those returned in the callback SVC:SELECT:LOCATION:.) The file is a DOS or UNIX text file, as appropriate to the host OS. If an error occurs, the file ends prematurely with the following line: error n.

save charpos option fn

Save character position information

Ⓞ € ¶

Same as save testpos above, except it saves the positioning of each character instead of each word.

scale all

SwiftView scales new pages so that the entire page will be visible, but does not draw them.

scroll [up| down| left| right] [n| default] Scrolls a given amount

These commands scroll a percentage of the view window size. The percentage to scroll is *n* with a value between 0 and < 100, or **default = 95%**. If the edge of the document becomes visible, it is placed at the edge of the window. For scroll up/down, if the top/bottom of the document is already visible and a previous/next page exists, **lpage previous**; **draw bottom/lpage next**; **draw top** commands are performed. For scroll left/right, if the left/right of the document is already visible, nothing is done.

select cancel Cancel selection

Cancels any outstanding selection.

select copy Copy selection

Copies the current selection, if any, to the clipboard. If there is no selection, the command is silently ignored. (Note: currently, the selection must be on the currently displayed page.)

select [rrect| drect] x0 y0 x1 y1 rect [enable| disable] Select and highlight a rectangle

search [rrect| drect] x0 y0 x1 y1 rect [enable| disable] Search a defined rectangle

When selecting a given rectangle, any selectable text in the rectangle is also selected. The **select** command highlights the text; **search** does not. If **rect** is enabled, a box is drawn around the selected rectangle. **Rect** defaults to **enable** for **select**, **disable** for **search**. The **search** command serves to initialize the selection for text searching. The selection remains until it is replaced or canceled, a **zoom selectrect** command is issued, or a new document is loaded by an **ldoc** command. In particular, it remains even if its page is no longer displayed.

Select/search **rrect** is similar to **select/search drect**, except the coordinates are drawing coordinates. The **rrect** and **rect** options are available in SwiftView version 5.2.0 and higher.

select string str matchcase [enable| disable] regexp [enable| disable] from [firstpage| currentpage| currentpageonly] found| foundifcurrent] Select next matching string

search string str matchcase [enable| disable] regexp [enable| disable] from [firstpage| currentpage| currentpageonly] found| foundifcurrent] Search for next matching string

select string selects and highlights the next string that matches **str**, changing the current page if needed to display the start of the string. The search begins at the character after the start of the current selection, if any, otherwise at the top of the current page, and proceeds down the page and on subsequent pages. If the end of the document is reached, the search wraps around to the start of the document. If no match is found in the entire document the search stops with a dialog message and any current selection is unchanged.

search string is like **select string**, except it does not use or modify the current selection or highlight the found string. Like **select string**, it changes the current page and view to display the found string. **NOTE:** This is especially useful for hyperlinking to a text string, where you don't want to highlight the linked-to text.

The **matchcase** parameter is optional. If **matchcase** is disabled (the default), any combination of upper and lower case letters is accepted. Any whitespace in the search string matches any whitespace in the text. If no new match is found, an error message is generated.

When `regexp` is yes, the search string is evaluated as a POSIX extended regular expression. See the **GNU regex** manual (http://www.math.utah.edu/docs/info/regex_1.html) for a description of the basic syntax. However, see **man 7 regex** on Linux, **man -s 5 regex** on Solaris, or **man 5 regexp** on HP-UX for how POSIX extended rules differ from GNU/POSIX basic rules. Here is a summary of the differences:

- no backslash in front of "{ } | " special characters
- special characters "*+ ?^\${}" are special anywhere in the RE
- add POSIX character classes, e. g., "[upper:]"
- add POSIX "bound"s, e. g., "{1, 5}"
- '.' matches newline.
- some minor differences in handling bogus expressions.

No internationalization support is provided (`setlocale()`).

The `from` option determines where in the document the search begins:

- `firstpage`: top of first page
- `currentpage`: top of current page
- `currentpageonly`: top of current page
- `found`: after the current selection/last found string if exists, else top of current page
- `foundifcurrent`: after the current selection/last found string if on current page, else top of current page

The default options are `regexp disable matchcase disable from found`.

select next Select the next occurrence of previous string

search next Search for the next occurrence of previous string

`select next` and `search next` move to the next occurrence of the string specified in the previous `select` string or `search` string command, respectively, including all options except `from`. The search begins at the character after the start of the current selection, if any, otherwise at the top of the current page. NOTE: `search` and the `regexp` and `from` options are new in SwiftView versions 5.0 and higher.

set allpensblack [enable| disable] Sets all HPGL pens to black

This command is used to force all HPGL pens to black, despite any settings inside the HPGL file. This is needed to display HPGL files that draw using white pens (hpgl command `SP0`), or when printing files with light-colored pens on monochrome devices.

set automarkup onpage [enable| disable] Enable/disable automatic hotspots on text strings.

This command enables automatic generation of special hotspots for URLs displayed in selectable text in the PCL or HPGL document. Enabled is the default. A URL is any selectable text (i.e., ASCII text, not vectors or images of text) string that begins with `http://`, `https://` or `ftp://`, up to the next whitespace or certain non-URL punctuation. The text is highlighted and underlined in blue and the hotspot commands the default browser to view that URL.

It is possible to configure all aspects of the URL hotspots and generate custom markups for arbitrary string patterns; regular expression definitions are supported. Contact tech support for more details.

set [bitonalgamma| dgamma] absval g Set scale to gray gamma(bitonal) or color display gamma(dgamma) or value - absolute

This sets the display gamma curve for "scale to gray" of bi-tonal documents and color images. The number may be between 0.01 and 100.0 and the default value is 1.0. Numbers higher than 1.0 will make the scaled bi-tonal image appear darker and the color image appear lighter. Numbers less than 1.0 will make the scaled bi-tonal image appear lighter and the color image to appear darker. A large bi-tonal gamma number is equivalent to `set bitonalscaling bestquality`. This affects printing as well as screen display.

For color images, this and the next 2 commands below apply gamma values on the values already implicit in a given file format. For example, TIFF color files have an "implicit gamma" of 2.2, requiring viewing programs to apply a gamma of 2.2 before viewing. This setting and the ones below are in addition to that implicit value.

set [bitonalgamma] dgamma] relval v Set scale to gray gamma or color display gamma value - relative

Sets value of gamma curve using a relative value from 1-100. The meaning of this value is defined by set ... relrange below.

set [bitonalgamma] dgamma] relrange r Set scale to gray gamma or color display gamma value – relative range

Sets the relative range value. The default **relrange r** value is 2.5, causing a **relval** of 100 to set gamma to 2.5. A gamma of 1.0 (no change) always corresponds to a **relval** of 50. Values larger than 50 make bi-tonal images darker and the color images lighter. Values less than 50 make bi-tonal images lighter and color images darker. Using a "-" on the **r** value inverts the sense of the **relval**.

set bitonalscaling [bestquality] bestspeed] false] set bitonalscaling

False is default and provides highest quality using scale to gray on monitors with at least 256 levels of color or gray. **Bestquality** also does scale to gray but is then thresholded to bi-tonal. **False** and **bestquality** have the same effect on monitors with less than 256 levels of color or gray. Both **false** and **bestquality** are affected by **set bitonalgamma**. **Bestspeed** scales faster but at the expense of poorer quality. It is also unaffected by **set bitonalgamma**.

set browser service Specifies the DDE service name of browser to which browse < URL> is sent

Supported DDE service names are "iexplore" or "netscape", maximum 40 characters. The default is "netscape". On UNIX, this command has no effect; only browsing with Netscape is supported. (The plugin sets this at SwiftView's initialization, depending on which browser it is running under.)

set cachebrowser [on] off] Specifies if browser caching is allowed or disallowed

set cachepcl [readonly] readwrite| memoryonly| none] Specifies file caching procedure

€

This command is only available with a SwiftExtract license. It is used to greatly increase performance when making multiple extraction/conversions from a single large file. **readonly** can read but not write cached files while **readwrite** can read and write cached files. The old command **set cachepcl on** is mapped to **set cachepcl readwrite**. The old command **set cachepcl off** is mapped to **memoryonly**. The command **none** specifies no file caching which is required to open files over 2gig.

set cachesize [NNNN] default] Set image cache size in KBytes

This command sets the maximum number of Kbytes of data cached in SwiftView memory, both for documents read as files and documents obtained from a SwiftServe image server. The default cache size is approximately 350 KB and **default** restores the cache to that size. **Cachesize 47** entirely disables caching and numbers up to 1000-2000 (1-2 MB) may be desired under some configurations. For good performance, **cachesize** should be at least the size of the largest TIFF page or PCL document to be viewed. If the limit is exceeded, the least recently used data is discarded from the cache as needed to make room for new data. Cached data is kept in memory in all cases and is lost when SwiftView exits. HPGL files are not cached in this manner, although the equivalent is done in other ways.

Set `cachesize` additionally sets the maximum number of PCL pages of searchable text kept in memory to `cachesize *3/10000`, or 105 pages by default. This default typically consumes about 1 megabyte of memory.

When using versions of SwiftServe before Ver. 1.2, caching is not available and the entire page must be downloaded to the client before any part of the image is displayed.

set cacheblocksize [NNNN] default] Set cache block size in bytes

This command sets the size of reads (of both local and image server files) in bytes; in doing so, it resets the default value, 2048 bytes for Windows and 24000 bytes for UNIX. The Windows default is lower due to more common use on slower systems and over dial-up TCP/IP links. Higher values increase performance by reducing the total number of client/server transactions. Lower values allow viewing operations to be interrupted more often and more quickly. For example, the entire first `cacheblock` is read before image display starts and making the block size smaller decreases the time before the first information is displayed. Values less than 512 bytes are not recommended. If Windows SwiftView is being used on a LAN, we recommend setting the block size to 24000.

set cachereset [enable| disable] Enable or disable closing of all files at an ldoc.

The default is `enable`. Closing files at `ldoc` does not discard the file contents from cache, but does ensure that the file modification time will be checked to insure cache consistency.

set checktext [enable| disable] Turn off check for bad fonts in Windows print files

SwiftView uses a workaround in order to allow searching and copying of text in print files from Windows 98SE, and to warn about unusable text in print files from NT/2000. This workaround can sometimes cause problems with PCL files not created with Windows printer drivers. Set `checktext disable` will suppress the workaround and warning, allowing these files to be searched and copied. The default value for `checktext` is `enable`.

set colorpalette [nochange| sysdefault] Set color palette use

The default SwiftView operation `nochange` means use the current color palette without changing it. This minimizes interference with other applications. If another application has set the palette to values optimized for its purposes, SwiftView's rendering of color images may suffer. `Sysdefault` causes SwiftView to change the color palette back to a "reasonable system default." This palette is acceptable for rendering color photos and should be adequate for other applications.

set count001pages [enable| disable| pattern] Effects how SwiftView processes x.001 files

If enabled, when viewing pages that end with `x.001`, SwiftView will stop after page 1 and look for a document with the same name and a suffix of `x.002`. This provides the ability to view multiple files as if all in one document. In some installations the suffix `x.001` is used for other purposes. Set `count001pages disable` turns off the default behavior of SwiftView for viewing `x.001` files. The default value is `enable`. Set `count001pages pattern` takes the first occurrence of "01" in a filename, plus any proceeding additional zeros, is replaced by the desired page number, with the same number of digits as the entire "[0]01" pattern. The first page of the file is loaded on an `lpage/dpage`, and page counting proceeds until a matching numbered file is not found. If the page number exceeds the number of digits in the pattern, additional digits are added. Examples:

```
set count001pages pattern| ldoc my0001.tif ->
my0001.tif, my0002.tif...my9999.tif, my10000.tif, my10001.tif...
```

set count001pages pattern| ldoc mydir01/my.pcl ->
mydir01/my.pcl, mydir02/my.pcl...mydir99/my.pcl, mydir100/my.pcl...

set count001pages pattern| ldoc mytif.001 ->
mytif.001, mytif.002...mytif.999, mytif.1000, mytif.1001...

set dialogs [enable| disable] Enable or disable dialog box display.

This command disables or enables display of all incidental dialog boxes on MS Windows systems. This enables use of `sview -c` ..." without seeing any dialogs. Directly called dialog boxes (`gui dialog ...` commands) are not affected.

set displayautorotate [enable| disable] Enable or disable autorotation of PCL pages

To make the most text readable, this command enables or disables automatically rotating each PCL page on display. Display rotation does not affect printing, but does affect save to file. The default is `enable`.

set docinfo [Title| Author| Subject| Keywords| Creator| Producer| CreationDate| ModDate] Specifies PDF document information.

set docname fn Set current document name override to "fn"

This command sets the name of the current document file, for `$docname/$docdir` and reload purposes, overriding the normal setting from `ldoc`. To restore normal settings, set `docname` to an empty string, " ".

set dps [enable| disable] Enable or disable Display Postscript

Disable or enable use of Display Postscript on X Window displays. This has an effect only when DPS is available. Enable is the default.

set displaycolor [color| grey| gray| bitonal| enable| disable] Set color display

P ☺

The same as `set printcolor`, for display. With bitonal documents or in Standard or Lite SwiftView, all non-bitonal settings have the same effect, limiting display to grayscaling, as there is no color display allowed. Note that `tiff/jpeg/pcl6/hppl` can display in color in Standard SwiftView.

set env str Set environment with string str

The environment variable is set inside SwiftView only and is lost when it exits. Environment variables can be inserted in ICS commands, using the macro mechanism discussed under "**Environment Variables and ICS Macros**" in the Configuration Manual. For example, we could set the environment variable `NDGFONT` as follows:

```
set env "NDGFONT= C:\applications\fonts"
```

NOTE: no trailing backslash is allowed.

set filename fn Set default output file name

☺ €

This is the file name used when printing to file and when names are not specified in a `save` command. This is also the file name presented to and modified by the user on the output menu.

set filetype [pcl| hpgl| tiff| unixtext| dostext| pdf| default] Set how to treat ambiguous files

If a document is not clearly identified as something other than PCL, HPGL or plain text, treat it as this file type. Many PCL and HPGL files do not explicitly state the file type. By default, if a file is not clearly identified, SwiftView takes its best guess at an HP type based on the file suffix (.hpg, .plt, .hgl = HPGL, .pcl = PCL), or else on analyzing the beginning of the file. With unusual text/PCL files (e.g., those documenting HPGL commands), it may not get the correct answer, or give up and default to PCL/DOS text. These commands tell SwiftView which file type to assume in the condition that the file could be either HPGL or PCL.

This command is useful when opening files without extensions, or when receiving piped data as input.

set fractionallinefeed pct Set the minimum vertical text advance interpreted as a linefeed

A tuning parameter to set the minimum vertical text advance that will be interpreted as a linefeed when extracting plain text to the clipboard or file, where **pct** is a positive real value. The default value of 1.0 matches the old behavior. A value of 0.00001 insures that all text is ordered by any y advance, and that any y advance generates a linefeed in extracted text. A value in the 0.00001 to 0.6 range is recommended for landscape tables without any subscripts. Normal text documents with very low subscripts may require a larger value, such as 1.5.

set fractionalspace n Set the distance at which words are divided

Some files have large micro-spaces between the characters of a word that SwiftView can incorrectly convert to a space when copying or saving plain text. This command lets the user control the micro position width treated as a space when extracting plain text. It answers the question, "How big must a gap between characters be before we can assume it is a space character?"

If the micro-space is more than this value times the font's default space width, but less than 1+ this value times the space width, a single space is generated. Larger micro-spaces are converted to enough spaces to place the next character at its actual distance from the left margin to the nearest font space width. The default value for **n** is 0.2.

set invertimage [true| false] Invert the incoming bitonal or gray scale image

The value **true** inverts the display polarity (white on black vs. black on white) of both bi-tonal and gray scale images. **False** preserves the original and is the default.

set jbig2_maxcomp n [10-20000] Sets the maximum number of elements to be cached during JBIG2 file conversions to PDF

set jbig2_weight_factor d[0.0-1.0] Sets JBIG2 weight ratios

This command affects JBIG2 compression when used with conversion to the PDF file format. Default is .8

set [ldocinit| dpageinit| draw| drawtop| drawbottom "..."] Set default viewing behavior

Set **ldocinit** determines the ICS commands to be performed automatically after an **ldoc** command. The default is **rotate 0 | scale default**. Note that non-markup commands may not follow **onpage...markup** when defined in **set ldocinit** commands; they will not get processed until after the **onpage...markup** is processed. For Windows, if **set ldocinit** is not defined by the user, **set ldocinit** is defined to **rotate 0 | scale default | printer number 99 type MS_BEST command none alias "cur, cur, cur"** to ensure that CNTL-P printing will function regardless of file type specific settings from the previous print menu print

job. For UNIX, the default definition is `rotate 0| scale default`, as printers must be defined by the user. Note that if you send `set ldocinit` from a web site, there are problems supporting both UNIX and Windows SwiftView.

`Set dpageinit` determines the ICS commands to be performed automatically after a `dpage` command. The `dpage` command is sent on all GUI page selection operations. The default is `draw last`.

`Set draw`, `set drawtop`, and `set drawbottom` determine the actual ICS commands to be performed on `draw`, `draw top`, and `draw bottom` commands. The defaults are `draw default`, `draw lasttop`, and `draw lastbottom`, respectively. `Draw` is sent on a `rotate` command or when a document is opened from the user interface. `Draw top` and `draw bottom` are sent when scroll keys change pages.

set maxjbigobjects [1-1000000] Sets the number of elements to be cached during file conversion to JBIG2 PDF

Default is 100000.

set nofileerror [enable| disable] Set if file dialog is displayed

This command enables or disables the "file not found" error to be displayed when a requested file cannot be found. The default is `enable`.

set onexit <cmd> This command will be executed as the last action before SwiftView exits.

The ICS command supplied by this command will be executed as the last action before SwiftView exits. It can contain any ICS commands, pipe-separated, except the only dialog allowed is "message". The commands will be fully processed, except for page counting.

set output [enable| disable] Disable document printing

The default is `enable`. A false value disables all printing, plotting, and save output (`plot` and `save` commands), putting up a message instead: "printing or saving this document is not allowed." Note that print buttons or key assignments are not changed (i.e., give the message above when pressed). These can be completely deactivated using separate ICS commands.

set pagespertick [1| default] Sets the callback output for page counting

This command forces SwiftView to report page information in the callbacks on every PCL or HPGL page while counting. The default is to count every 5 pages.

set pdfconfig <filename> Sets the extended PDF configuration file

set pdfhintflags [enable | disable] Improves viewing quality on PDF files

This command helps improve rendering issues with some PDF files.

set pdfpassword str Apply this password for PDF files

Allows user to set a password for PDF files that require one. If `str` contains the password, the user will not be queried for one; however, if the password is incorrect, a dialog is displayed prompting for a password. To disable the incorrect password prompt see `set pdfpwfailoption`.

set pdfpwfailoption [enable | disable] Bypass PDF password request

Setting this command to disable will force SwiftView to quit when a PDF is supplied a bad password. If the command is set to enable SwiftView will behave normally and wait for a password to be provided. Recommended for unattended batch conversion. Enable is the default state.

set pdfres n [60-1200] Output PDF quality

Command must be specified before the file is loaded. This only affects the quality of output PDF files from PDF input files. Default is set to 600.

set printarea [paper| printable] Choose how much of the paper to use

Determines the output print area used by `set printscale fit`, `set printxposition`, and `set printyposition`. `Paper` is the default, and means use the entire paper size. `Printable` means use the printable area reported by the printer driver.

Many printers have small non-printable margins, which are normally not a problem because most documents have margins covering the non-printable area. But if the document has data in the non-printable area, this data will be lost. In this case, use `set printscale fit` and `set printarea printable` to make SwiftView scale the original drawing to the printer's printable area.

set printautorotate [enable| disable] Disable automatic page alignment at print

When `printautorotate` is enabled (the default setting), SwiftView automatically rotates the drawing orientation during printing, aligning the long axis of the drawing with the long axis of the paper. This is what most users expect when printing. If the `printautorotate` is disabled, SwiftView matches the print rotation with the current display rotation. If, for example, the user rotates the drawing 90 degrees on the display, SwiftView rotates the drawing 90 degrees during print, allowing the user control of the print rotation.

set printcolor [color| grey| gray| bitonal| enable| disable] Set color output

P C

Sets the maximum print output color format. If set to "gray" or "bitonal" it will cause color or grayscale documents to print gray or bitonal (gray is not implemented in 7.2). The default for Pro is color. Enable is a synonym for color, disable for bitonal.

Set `printarea printable` does not affect clipping to the printable area; such clipping is a function only of the printer and printer driver.

This command does not affect output to TIFF, CALS, PCX, or direct printing.

set printcompressformat [beststd| std| unc| g3| g4| faxlo| faxhi| packbits| jbig2| jpeg| flate] Set compression format for saving

P C

This command sets the image compression method for save and plot drivers that support multiple compression formats (TIFF and PDF). Note that only **uncompressed, packbits, jpeg and flate** support color.

beststd Default setting. Uses the best, widely-supported standard compression format supported by the driver. Tiff compression is performed on a per-document-page basis, and a per-entity basis in PDF. The actual formats used are shown in the table below.

<u>Format</u>	<u>Available Compressions</u>	<u>Beststd: Color</u>	<u>Beststd: Bitonal</u>
TIFF	unc g3 g4 faxlo faxhi packbits	packbits	g4
PDF	g4 flate jbig2	flate	jbig2

std The same as "beststd", but uses g4 for bitonal images in PDF, insuring compatability with older software (e.g. Acrobat 5.0) that does not support jbig2. (PDF always uses flate for color. It uses jbig2 for bitonal unless "std" or "g4" is selected.)

If a requested format is not supported by the driver, it uses the "next best" format it does support. Color formats are only substituted for color, bitonal for bitonal.

set printcompresslevel N Set the compression level for various save outputs

This command tunes the degree of compression of the flate, JBIG2, and JPEG compression methods where used in PDF, PNG, and TIFF. Value is from 0 to 100. Generally larger values = smaller files and longer compression times, but the exact results are driver-dependent. Compressions above 50 may result in increasing degradation of image quality (JBIG2/JPEG), becoming extreme above 80 (JBIG2). 0 means uncompressed if supported (not supported in JBIG2). The default of 50 is a good compromise for all formats.

Printcompresslevel controls a JBIG2 "% match" setting, whose square is approximately the percentage of the pixels of two glyphs that must match to be coalesced into one glyph. The setting does not significantly affect speed. Printcompresslevels 0-20 are mapped to a JBIG2 match level of 100%, 20-80 to 100%-95%, and 80-100 to 95%-0%. Printcompresslevels of 50 or less should only cause slightly different renderings of the same glyph to be coalesced. Levels above 50 can result in occasional erroneous characters (e.g "I" for "l"); levels above 80 will render text files unreadable.

set printduplex [disable| long| short| original] Sets the output files duplex command

This command allows forcing duplexing in the outputted file, regardless of the files own duplex commands. Original specifies to follow the documents own duplex commands and not force any. Disable forces simplex on all output. The default is "set printduplex original".

set printinray[N] original| disable] Specifies, passes, or disables the input tray

Requests the specified input tray N for each page, the tray ID specified in the original PCL file, or passes no tray commands at all. Currently this command is not sent by the print dialogs and the default is off. Replaces the older "pcl passtray" command.

set printinraymap originaltray N tray N Remap document tray commands

Remaps the original document tray commands for the Windows Print and Fast Print methods. Has no effect unless there are tray commands in the original document and "set printinray original" is given.

set printinraymap originalsize name tray N Remap document tray commands
on paper size

set printinraymap originalsize pcl N tray N

set printinraymap originalsize clear

Request a tray based on the page size. Replaces/overrides the older "pcl *out_tray" commands. Overrides "set printinray" and "set printinraymap". Clear erases all printinraymap originalsize maps.

set printinraywinmap pclid windriverid Remap document tray commands for Windows
print

Equivalent to the older "set printwinray" command. Applied to the resolved printinray, if any, for Windows driver output only.

set printjobname [filename] default] Specifies the print job name sent to the spooler

The default print job name is set to the filename of the viewed file. This command allows specifying a different print job name for the output file, or setting it back to the default. This is done by adding a PJLCOMMENT tag in the output file, which will be understood by both the Windows print system, as well as the front panel of most printers.

set printmenudefault method [fast| direct] [enable| disable] Prevents defaulting of a
method

This prevents a print method from being selected as the default. **Enable** is the default. It does not prevent the user from selecting a method, but does restrict the MS_BEST method default selection (immediately affecting control-P printing). Note that Windows printing cannot be disabled, as it is the only possible method for some cases.

set printmenu printtofile [off | on] Prevents printing to file

Disables the print-to-file checkbox and printing to FILE: printers, regardless of licensing. Default is on. This will also disable all batch printing to FILE: printers as well.

**set printmenudefault printertype [pclps| notp] passpapersize [y| n] Set default scaling
scale [1| all| printable] color [enable| disable] res [original| n] duplex
[long| original| disable]**

These commands set the default universal print dialog options on a per-printer-type basis, meaning that PCL & PS printers can have parameters set one way, and all other printer types can be set another way. Defaults are used when the user has not changed options or has reset them to defaults with the Reset button on the options dialog. The first parameter selects the printer type that the subsequent options will apply to. **Printertype** is the analyzed printer type used to set the print method.

pclps	command will apply to printers that are pcl5 monochrome, pcl6, or postscript.
notp	command applies to all other types of printers. These options do not affect direct printing or cuts.
passpapersize y	paper size used to print is that of the digital file being printed and sets scale to 1.
passpapersize n	paper size used to print is that specified in the Printer Properties.
scale 1	no enlarging or reduction will happen upon printing.
scale all	when printing, the document will be scaled up or down to fit all of the paper.
scale printable	printing will be scaled up or down to fit only the printable area of the paper.
color enable	enables color output P C

color disable	disables color output P C
res original	sets output resolution to that of the original file
res n	sets output resolution to n P C
duplex long	sets the duplex option to long edge binding
duplex disable	sets the duplex option to simplex
duplex original	sets the duplex option to obey document duplexing

Initial settings are:

```
set printmenudefault printertype pclps passpapersize y
set printmenudefault printertype pclps scale 1
set printmenudefault printertype pclps color disable
set printmenudefault printertype pclps res original
set printmenudefault printertype notp passpapersize y
set printmenudefault printertype notp scale 1
set printmenudefault printertype notp color disable
set printmenudefault printertype notp res original
```

These options do not affect direct printing or cuts, and do not affect MS_BEST and control-P printing until the user actually prints from the universal print dialog (i.e., until the dialog actually issues `set print xxx` and `pcl passpapersize` commands).

set printouttray [N] original| off]

Defines output tray

Allow control of print output tray for PCL and PostScript drivers. Requests the specified output tray ID on each page, or a tray ID derived from the original PCL file. Currently this command is not sent by the print dialogs. Default is off.

New plot/save range option: `oN` = request output tray N, `N >= 1`.

SwiftView recognizes PCL output bin and PDL OUTBIN commands in PCL files, passing the PCL output bin number through as-is and mapping PDL OUTBIN to PCL/Postscript output bin commands: UPPER to 1, LOWER to 2, and "OPTIONALOUTPUTBINn" to n. On the first page only of a PCL output file, any tray setting is duplicated with a PDL OUTBIN command. Other PDL OUTBIN values may exist but are not well documented and are ignored for now.

set printpassword s

Defines password for output file

Sets the password used to encrypt saved output, e.g. PDF. Requires the installation of the SwiftView PDF Encryption Utility which is currently available for Windows only.

http://www.swiftview.com/product/current/svinstall_pdfencrypt.exe

set printpjl [enable| disable]

Permits the inclusion of built-in PDL commands in PCL5 print driver output

SwiftView will currently insert standard PDL commands in the header of PCL files when printed or saved in certain conditions. This command allows disabling the addition of any PDL commands in SwiftView's PCL5 output. The default is enable.

set printprerotate [enable| disable]**Enables prerotation of landscape pages**

Enable prerotation of landscape pages when converting to drivers which support driver rotation. "disable" is the default, and enhances conversion speed. "enable" forces SwiftView to perform rotation itself instead of depending on the output driver. This may be necessary if a PDF reader is slow to rotate.

set printquality [best | speed | default]**Set printer output type**

Allow SwiftView to choose appropriate quality settings, implies "set printbitonalscale default" Currently, this affects PDF in the following way: When printquality is set to "default," anti-aliasing will be disabled for PDF printing if the target resolution is 300 dpi or higher, otherwise it will be enabled. If printquality is set to "best," anti-aliasing is always enabled, and if set to "speed," anti-aliasing is never enabled.

set printres [n] original]**Set output DPI resolution**

This command sets the DPI for the FAST and WINDOWS print methods, as well as the save resolution, to n for both x and y. "original" causes plot and save commands to output at the resolution of each original document page. If the original document format does not specify a resolution, 300 dpi is used. For pcl5 files the resolution determined by "pcl [force|default]res" is used.

set printscale [auto| fit| n]**Set print scaling**

Print scaling determines how the plot command will scale the document or the cut area of the document to the output page. This command will not affect the output if the Universal Print Dialog is used. Note that the universal print dialog always issues set printscale, set printarea paper, and pcl passpapersize commands. Sending one of these commands does not affect printing with the dialog, but dialog settings will affect subsequent custom-configured plot commands or the plot 99 all issued on control-p. fit will fit the document or cut area to the page; n is a positive real number value that applies a fixed scale factor. The default is "auto", which does the following:

For plot cut, rcut, or view: scale to fit the paper or printable area, per set printarea.

For full-page printing:

1. Ignore set printxposition, set printyposition, and set printarea; scale to fit is to printable area only, and all scale factors are the same in x and y.
2. If NDG_PAS is set to a number between 0.0 and 1.0, the drawing is scaled by that factor (i.e., scaled down), centered on the page.
3. For PCL or HPGL documents, attempt to prevent any data from being clipped off by the printable area. If the document is larger than the printable area by less than 0.6 inches in either axis, attempt to position the document so that no non-background pixels are clipped off at 1:1 scale. If this is not possible, scale the non-background pixels to fit. The resulting scale factor is decreased by 0.997.
4. Or if the document is not HPGL/PCL, or is equal to or smaller than the printable area or larger by 0.6 inches or more, in both axes, scale 1.0 if in both x and y the document is larger than the printable area, less than 0.1 inch larger than the output paper size, and the printable area is less than 0.55 inches smaller than the output paper size, in both axis. Note that this may clip off some data in the margins.
5. Otherwise, scale to fit * 0.997 if the output paper size is less than both 8.7" wide and 11.9" high.
6. Or scale 1.0, centered on the paper, if the document is smaller than the printable area.
7. Or scale to fit * 0.997.

This command does not affect plot to TF*, PCX, or DIRECT, which always print 1:1 for whole pages and scale to fit printable area for cut/rcut/view (but DIRECT ignores cuts). `auto` is the same as the pre-SwiftView version 5.5 operation, except that cuts always scaled to fit printable area, and the fixed 0.997 scale factor to insure that data exactly at the edge of the drawing is not clipped used to be applied in all cases.

set printsize [pcl n] name s| width x height y| source original| source windriver] xmargin x ymargin y orientation o oversize n undersize m Manual specification of output paper size

This command allows you to manually specify the size of an output file for printing and saving.

pcl n	n is PCL escape sequence number specification for paper size.
name s	Specify output size by name listed in Appendix B: Paper Size to PCL ID Map
width x height y	Allows custom size specification in inches/cm depending on how units are set.
source original	Specifies the output file to be the same size as the input files size.
source windriver	Specifies the output file to be the size set in the Windows properties for the driver.
xmargin x ymargin y	Allows adjustment of output files margins.
orientation o	Specify output file orientation, options are landscape and portrait. The default is portrait.
oversize n undersize m	Specify how much a standard PCL page size can be greater or less than the original document's width and height paper size for SwiftView to output at exactly that standard size, rather than the exact original size. The defaults for oversize and undersize are .02" for PCL original documents, else 0.4", and are restored with a negative setting. The minimum value is .02.

set printspotcolor [enable| disable] Enables/disables spot color compression

Optimizes compression in PDF and PCL output which normally results in smaller files. The output is divided into small squares, and color image data is output only for those squares that actually contain gray or color data. Default is enable.

set printtext [enable| disable] Set inclusion of searchable text in output

P ☺

Sets the inclusion of searchable text in the output of the PCL5 and PDF. The inclusion of searchable text allows the documents to be searched, but may also increase the file size slightly. The default is enable.

set printtiff n Set Tiff output fill order

P ☺

Some older fax systems can only accept TIFF files with a specific fill order. Default is 1 for Gammafax card compatibility. The other available option is 2.

set printwintray pclid winid Remaps pcl input paper source (tray) ids to Windows driver "DEVMODE" source ids

Many Windows drivers use non-standard DEVMODE tray ids (>256), so SwiftView's default tray mappings for Windows print may not get the desired tray. You can map your PCL file tray ids to different driver-specific values with "set printwintray pclid winid".

pclid PCL tray number used in PCL escape sequence
winid Driver DEVMODE tray ID

Contact Tech Support for assistance in determining the correct tray ID's for your driver and printer.

set printxposition [center| left| right] Set printer positioning offset

set printyposition [center| top| bottom]

set [printxoffset| printyoffset] n

These commands control the position of the image on the printed output when the document or cut area does not exactly fit the print area. Positions are relative to the output paper in its "normal" orientation. Left, right, top, and bottom align the adjacent edge of the image to the given edge of the print area. Printx/yoffset adds an additional position offset in drawing units; positive values shift the document toward the top/right of the printed page. The default is printx/yposition center and printx/yoffset 0, if set printscale is not auto. These commands do not affect output to TIFF, CALS, or PCX, or direct print.

NOTE: if you specify a non-centered position with printautorotate enabled, which of two edges of the image is placed at the edge of the paper is undefined.

set product {product_name} RLM Command to load/re-load a license

The new licensing system supports a model which allows multiple license files to exist on both local systems as well as a licensed server. When a license is loaded the application then writes the product_name to the users Application Data folder and sview.ini file. If the license fails to load, the new product_name will be ignored.

This command is only useable in Windows.

set ps2output [recompress| default] Set G4 output mode for Postscript Level 2 printing

SwiftView prints bi-tonal images very fast on Postscript Level 2 printers by sending image data to the printer encoded in CCITT Group 4 compression format. The default mode converts all TIFF and PCL (but not HPGL or JPEG) source information to Group 4 bi-tonal images. Each 4 binary bytes are, in turn, encoded as 5 displayable characters, because many PS drivers work best with displayable information. If the original file format contains CCITT Group 4 data (TIFF or CALS files), that data is not decompressed, but rather just PS encoded and sent to the printer. The result in that case is very low host processor load.

The recompress mode works much like default except that source Group 4 images are first decompressed and recompressed before the Postscript encoding. SwiftView is capable of reading many G4 images that Postscript will not accept. This enables such problem files to be printed with minimum size print jobs at the expense of additional host cycles. As a result, **this option should not be used except when such problem G4 images are encountered.**

set quotechar 0xNN Set ICS command quote character

This enables a new quote character in addition to the standard double quote (") used in ICS commands. NN is an 8 bit hexadecimal value. A good character to use is 0x5e (^). This simplifies using ICS command lines from other programs. The following characters are not allowed: space | , # % \. Either a new quote character or the standard quote character can be escaped in strings with ' \' so that char \^ is the actual character ^.

This command does not affect the command line containers `-ics" ..."` and `-c" ..."` because those are controlled by UNIX and Windows requirements.

set rlminit [file_path] Sets the RLM search path for license files

Set the rlm license folder path to the value specified. This setting overrides the default path for license loading. For Windows users, the default location for the license files is the `%ALLUSERSPROFILE%\SwiftView\folder`. For *NIX users, the search path defaults to the current working directory.

set roam [on | off] Sets the RLM control for roaming licenses

This command only applies if the user has a valid license with the `product_name` set to `rlm_roam`. Roaming only applies to Licenses hosted on a license server. Additionally, the licenses must have roaming enabled when they are generated.

Action When Called: Checks in the current license, enable roaming settings and checkout the same license. If the checked out license support roaming, check the license back in and check it out one more time. We write the roaming license value to the users `%APPDATA% svview.ini` file.

set savetextlpi n Set the lines per inch when saving to TEXT



This command allows adjustment of the lines per inch in SwiftConverts Text output. The default is

set select charset [unknown| EBCDIC] Set character encoding

This command sets an assumed character set for the document. Some documents use encoding other than ASCII. By default, text is assumed to be iso8859-1 (ASCII) for selection, search, copy, and save text. This is done regardless of which charset the document declares, i.e., files sometimes lie. Using this command allows selection, search, copy, and save text for documents with other character sets. EBCDIC is currently the only character set supported other than the default, "unknown," which is treated as ASCII.

set select [drawing] document] Set the text selection method

Sets the text selection mode. In drawing mode, only characters within the selection rectangle are selected. Document mode highlights the complete paragraph consisting of all words between the upper left corner of the selection rectangle and the lower right corner. The default is `document` mode, the only mode available in SwiftView versions 5.4.0 and lower.

set select [enable| disable| iftext] Set the select/zoom mouse user interface

This command determines the how the user makes a selection or zooms, as follows:

enable means an unmodified mouse button performs a selection, and "Alt"- left mouse zooms on release of the mouse button, without selecting anything.

disable is the default. It reverses the "enable" actions: an unmodified mouse action zooms and "Alt"- mouse enables selection.

iftext enables unmodified mouse select if the document is PCL or HPGL with selectable text.

set sie [enable| disable] Enable/disable use of NCD Simple Imaging Extensions

Controls the use of Network Computing Device' s Simple Imaging Extensions. This command works only when SIE is available: on NCD X terminals with sufficient data memory. Enable is the default.

set swiftservetimeout [t] default] Set timeout on sever file access

This sets the time in seconds that SwiftView waits for a file to be provided by SwiftServe once it has established a connection. The minimum value is 6 seconds and the maximum is about 8 hours. The default (restored by default) is 60 seconds. This is useful, for instance, when to increase the time allowed for the server to obtain images from a busy shared optical jukebox.

set sourceimagephoto [missing| override] v Replace missing / override existing TIFF photometric interpretation tag

These commands enable bad or missing TIFF photometric interpretation tags to be replaced at viewing time with the desired value v. Legal values for v are *WhitelsZero* and *BlacklsZero*. The default setting is `set sourceimagephoto missing WhitelsZero`. It is not legal for these tags to be missing, but using these commands enables SwiftView to, in some cases, support these "illegal" files.

set sourceimageres [missing| override] n Replace missing or override existing TIFF resolution tags

These commands enable bad or missing TIFF resolution tags to be replaced at viewing time with desired value n. Missing or wrong tags affect only printing and markup.

set textencoding [latin1 | utf8] Expands clipboard limit

By default, all text extracted from the document to text files (`"save text[pos]"` , callbacks (e.g. `SVC:SELECT:STRING:`) and the clipboard is limited to 8-bit. Any characters which required a two-byte Unicode character to represent them are converted to *rubout* (0x7f); *one-byte* (Latin-1) Unicode characters are passed through. If the PCL file encoding is known to SwiftView (i.e. is either in an internal mapping or in an installed "symmap" file), it is mapped to Unicode, so any PCL characters that map to Latin-1 are returned in Latin-1. Otherwise the code is passed through unchanged. The text is put on the clipboard as `CF_TEXT`. This generally only works for Latin-1 text in the PCL file.

"Set textencoding utf8" changes the encoding in text files and callbacks to *utf-8* (multibyte unicode), and text is placed on the clipboard as `CF_UNICODETEXT`, so text in any encoding known to SwiftView can be sent. New select option: `select string str code latin1|utf8`

set timedcommand repeat n ics Set up ICS commands to be repeated

Specifies that the given ICS commands will be executed after n seconds, or every n seconds if repeat is given. Only one `timedcommand` can be in effect; so any previously defined `timedcommand` is canceled. If n is zero, no `timedcommand` is set, so it acts as a cancel (a command string must still be given, but an empty string ("") is OK).

set zippassword str Automatically apply this password for zipped files

Allows user to set a password for zipped files that require one. This command has an alias, `set z` to obscure its purpose to the casual observer. If `str` contains the password, the user will not be queried for one; however, if the password is incorrect, an error is displayed. NOTE: this command applies only to `pkzip/infozip` files.

set zoomfactor n Define zoom in/out factor

Define the amount for zoom in and zoom out. Legal values for *n* are between 1.0 and 10.0. The default is 2.0.

system s Execute system call with string *s*

units [inches| centimeters] Set drawing units to inches or centimeters

This defines how dimensions are treated in all subsequent SwiftView non-markup ICS commands that use drawing units. Changing units has no effect on previously executed commands, only subsequent commands. The default is *inches*. This command does not apply to markups (see *markup units*) and does not affect *hpgl* pen widths.

use reset Reset all "onpage use" commands

Clears all *onpage use* commands, and all states associated with an *ldoc*. Equivalent to *onpage reset*", but without clearing any markups. It does the following:

- 1) clear *onpage use*
- 2) if *set cachereset* is enabled, *resetcache*
- 3) release document memory and file resources
- 4) set page count to 0/0
- 5) blank the displayed page
- 6) set the current document name to the outermost enclosing ICS filename, else none.

winsize xa ya ws xp yp Set display window size and position

Set window size to aspect ratio defined by *xa* (horizontal)/*ya* (vertical) components. The percent of largest screen dimension used is set by *ws* (.00-1.00). The window size is reset to this value every time a new document is loaded.

The optional parameters *xp* and *yp* determine the position of the window relative to the screen, and have values between -1.0 and +1.0. Positive values mean offset (in percent of screen) of the left/upper corner of the SwiftView window with respect to the left/upper corner of the screen. Negative values are for the lower right corner of the window with respect to the lower right portion of the screen. 0, 0 places the SwiftView window in the upper left corner of the screen. -0.01, 0.01 places the SwiftView window at approximately the lower right corner of the screen, with some variation due to window decoration thickness. This feature is functional for Motif 1.2+-compatible window managers only, as well as for MS Windows.

zoom [in| out] Zoom in or out by 2 about center of display

zoom [rrect| direct] x0 y0 x1 y1 Zoom to drawing or display rectangle

These commands zoom in (more magnification) or out by a fixed scale factor. Use *set zoomfactor* to change the *x2* factor for *zoom in* and *zoom out*. *Zoom rrect* and *zoom direct* specify what area of the current page to display using a rectangle in real drawing or pixel (display) coordinates, respectively. If the shape of the window does not match the shape of the rectangle, the extra screen area is filled in with additional parts of the page.

zoom selectrect Zoom selected rectangle to fit window

Zoom the selected rectangle to fit in the window, if a selection exists, or else execute *zoom in*.

Chapter 2: SwiftView Environment Variables

The following environment variables can be set by the user, either in the shell/DOS/Windows environment, or using the ICS command “set env”. These environment variables can be set for both UNIX and Windows systems, however some of the variables are used only in UNIX. See "Environment Variables and ICS Macros" in the SwiftView Configuration Manual to learn how to use environment variables in ICS commands, and for a list of environment variables that are automatically set by SwiftView rather than being set by the user.

NDGCONFIG Defines the directory where a configuration file (svstart.ics) unique to the current user resides

If \$NDGCONFIG/svstart.ics exists, it replaces (is not merged with) \$NDGUTIL/svstart.ics at run-time. This variable also affects finding the X resource file on UNIX systems. It does not affect finding PCL font files or the operation of ndgstart.

NDGTTFONT Defines the directory for True Type Fonts

By default, SwiftView's extra True Type fonts are assumed to be in \$NDGUTIL\ttf. . NDGTTFONT overrides this default location and can be defined as a website http: location, with all but standalone UNIX SwiftView.

NDGBMFONT Defines the directory for Bitmap Fonts

By default SwiftView's extra bitmap fonts are assumed to be in \$NDGUTIL\p300x300. NDGBMFONT overrides this default location and can be defined as a website http: location, with all but standalone UNIX SwiftView.

NDGICS0, NDGICS1, NDGICS2, ... Pre-defined ICS commands

These environment variables can be defined to contain ICS commands. They are loaded after svstart.ics, merging with or overriding the ICS commands defined in that file.

NDGHPGL Defines the model number of HPGL raster plotter with HP RTL extensions being used

The default is NDGHPGL= M250. The other supported values are M255, M650E (36" wide HP DesignJet), M650D (25" wide HP DesignJet). If your HP plotter is not compatible with these models, contact us for a procedure to make additional settings enabling your use of that plotter.

NDGMACRO Defines the directory for Macros

If not defined, the value of NDGMACRO is assumed to be \$NDGUTIL\macros. NDGMACRO can be defined as a website http: location, with all but standalone UNIX SwiftView. For more information, see the section "Macro Support" in the Swiftview Configuration Manual.

NDGNOCUSTOMSIZE Disables passing of CUSTOM paper size (101) in PCL output

If NDGNOCUSTOMSIZE is true, PCL size 0 (default) is output instead of 101, but the PCL PAPERWIDTH/HEIGHT still are output so SwiftView knows the size. Printers will print on the default paper without stopping.

NDGNOFILE Defines default page when a file can not be accessed

The name of the image file displayed when SwiftView is asked to display a file that it cannot access (instead of displaying a dialog box). Simply define a file of any supported type using this command.

NDGNOSIE Control use of NCD Simple Imaging Extensions

Defining this environmental variable as "TRUE" disables support of NCD's Simple Image Extensions (SIE). Otherwise SIE is used, if it is available (it is free of charge on all NCD X terminals) and enough memory is installed. We recommend 8MB of memory per terminal, but 4MB is sufficient on monochrome terminals if only a few windows are currently in use. SIE does not support "scale to gray" (like SwiftView host-based scaling) but off-loads more than 95% of host CPU cycles. Use the ICS command "set sie disable" for the same effect.

NDGNOCOLOR Force TIFF display at 8 bit gray scale

If defined as "TRUE", this makes color TIFF files display as 8-bit gray scale on color terminals. This may be desirable if other applications have already allocated all available colors for their own use.

NDGONEPRINTJOB Define output on UNIX systems

If defined as "TRUE", this causes all UNIX print pages to be submitted under a single print command after all temporary print files are written. This requires enough system "temp" space to hold all pages prior to printing.

NDGDEBUG Turns on debugging for UNIX SwiftView

If defined as "TRUE", this places considerable additional information regarding startup, printing and other operations in the file \$HOME/.sviewerror.

NDGDEBUG_FILE Defines Debug file in UNIX SwiftView

Defined as the name of a file into which to write debug information will record considerable additional information regarding startup, printing and other operations on the display of the session or window which started SwiftView.

NDG_PAS Set scaling on UNIX SwiftView

NDG_PAS set to a number between 0.0 and 1.0 causes images smaller than the printer page size to "auto-scale" to the percent of the plottable paper size defined. That is, NDG_PAS=0.99 causes documents which otherwise would have been centered and scaled to fill 99% of the plottable paper. This will be superseded by an ICS command in a later release and anyone using it should be prepared for that change.

NOR5I18N Set Xlib Native Language for UNIX SwiftView

Using this variable causes SwiftView to set the locale (the native language) in Xlib and the operating system, in the standard X11R5/Motif 1.2 manner. Under certain operating system/locale combinations, this variable may have to be set or not set for proper localized operation. Regardless of the setting of NOR5I18N, the standard X11 Xt toolkit locale announcers (e.g., \$LANG) select the native language for all text strings displayed by SwiftView, by selecting which resource files are loaded.

SV_DISPLAY_RATIO

Adjust display ratio for reported aspect ratio

This variable allows adjustment of SwiftView's display aspect ratio. Some drivers report incorrect aspect ratios of the display, causing skewed images. This environment variable allows adjustment for this case. Values can be between 0.0 and 10.0. Setting SV_DISPLAY_RATIO=1.0 will fix most cases.

Chapter 3: ICS Callbacks

Each callback is a single newline-terminated line of text starting with "SVC:", with the exception of STAMPINFO, OUTPUT:CURRENT, ERROR, SELECT:STRING, SEARCH:STRING, and MESSAGE, which may be followed by additional lines not beginning with "SVC:". Generally, clients should not depend on the order in which different callbacks are received. Note carefully where spaces do or do not separate tokens.

SVC:BASELINE:ryorigin	Indicates the real y position of the text origin. Sent after the SVC:SELECT:LOCATION or SVC:SEARCH:LOCATION callbacks.
VC:COPYTEXT:str	Indicates the text string that was last copied
SVC:COVER:TRUE FALSE	Current page is is not a cover sheet. Generated by ICS "cover check"
SVC:DIALOGS:ENABLE DISABLE	Generated by ICS "set dialogs enable disable"
SVC:DOC:fn	Name of the non-ICS document (URL or file) requested by "ldoc fn" command. NOTE: this is not necessarily the "current document" (see the reload command), and is not sent for documents defined with "onpage use".
SVC:DOCINFO:filetype width height fn	The DOCINFO callback is sent at each page load. The filetype field contains a string such as "TIFF", "PCL", etc. Page width and height are in the current units (in/cm per the "units" command). Filename is the current document filename (see the definition of "current document" under the reload command). Filenames can contain spaces, so everything after the space following the height field is the filename. NOTE: SVC:PAGEDOC will report the current page's filename as defined by onpage, if any, while SVC:DOC: reports only non-ICS files loaded by the "ldoc" command. The order of these callbacks is not defined.

SVC:DUPLEX: [long short simplex]	Reports the duplexing status of the page
SVC:DVIEW:dx0 dy0 dx1 dy1 vx0 vy0 vx1 vy1 rescale	<p>1-4: drawing size in internal coordinates</p> <p>5-8: current view in internal coordinates</p> <p>The keyword "rescale" is optional. These coordinates are NOT normal ICS coordinates, and should not normally be used by an application programmer. The drawing size includes the gray border to control scrollbars, and coordinates are decipoints from lower-left to upper-right. Unless implementing scrollbars, use the DOCINFO, RVIEW, and WVIEW callbacks to track scaling.</p>
SVC:ERROR:str	Error. This callback can consist of multiple lines. The user needs to collect all lines until the next "SVC:."
SVC:FILENAME:fn	Current output filename, reported on a get filename command.
SVC:GUI:...command tokens...	Indicates which ICS GUI command was issued.
SVC:GUI:clientmessage str	Command/message for the client application to process. Indicates that a dialog box has been cancelled or closed by the user. Commands to GUI layer to initiate various input modes
SVC:IMPRESSION:0 m[+]	Reports the impression count m. Impressions are pages that actually contain data, and do not include blank sheets. + denotes the page count is not yet complete
SVC:MESSAGE:str	Message displayed using "gui dialog message" command. The command "set dialogs disable" will disable this callback as well.
SVC:NOTICE:str	Returned when general notices to user (e.g., reminder messages) appear onscreen. This callback is not disabled by the command "set dialogs disable", although the onscreen notices are.
SVC:OUTPUT:CURRENT:cp COUNT np LIST: printernum alias	Reports the number of the currently selected printer (cp), the number of defined printers, and a list of the defined printers from a "get plotters" command. This callback consists of multiple lines. Each line after the first gives the number of a printer and its alias string. The client application must collect all lines until the next "SVC:."

SVC:OUTPUT:START:str	Show "output in work" menu with "str. " The output process is about to begin.
SVC:OUTPUT:STOP	Hide "output in work" menu. The output process is complete.
SVC:PAGE:cp np	Current page state. Values cp = current page, np = number of pages. The number of pages terminates with a '+' character until page counting is complete. A callback is received for each page during the counting process.
SVC:PAGEDOC: filename	"Filename" is the URL or file from which the current page was obtained. This differs from SVC:DOC, and is non-null when documents are defined with "onpage use" (and SVC:DOC is not sent).
SVC:PAGEINFO: resolution n [bitonal gray color] compression [G4 JBIG2 JPEG FLATE]	Provides information on every page loaded in a document. Gives resolution, color, and compression information.
SVC:QUIT	Indicates when SwiftView has been closed.
SVC:RESIZE	Indicates that current draw window (not necessarily main window) has been resized.
SVC:ROTATE: n	n = current rotate state, with value: 0 = the normal state called for by the file 1 = rotated clockwise 90 degrees 2 = 180 degrees 3 = 270 degrees
SVC:RVIEW:xul yul xlr ylr	The current viewable region of the drawing in drawing coordinates

The following SVC:SELECT: or SVC:SEARCH: callbacks are sent on the select/search string/next/direct/rrect commands to indicate what text, if any, was selected, except that SVC:SELECT/SEARCH:SCROLL is sent only on select/search string/next:

SVC:SELECT:SCROLL:panval
SVC:SEARCH:SCROLL:panval The pan value, y that a select or search used to scroll the selected string so it was visible

SVC:SELECT:LOCATION:pnum rx1
ry1 rx2 ry2 str The page number, drawing coordinate bounding box, and text of a string segment found by a select or search

SVC:SEARCH:LOCATION:pnum rx1
ry1 rx2 ry2 str

SVC:SELECT:STRING:str The full string found by a select or search.

SVC:SEARCH:STRING:str

If "str" is empty, no match was found. An empty STRING followed by a non-empty STRING will be sent when wrapping around from the end to the beginning of the document. All LOCATION callbacks for a select or search are returned before the corresponding STRING callback. If two consecutive substring-bounding boxes mostly overlap each other, only the second is returned in a LOCATION; this permits xoring with the set of bounding boxes.

NOTE: in a user selection operation "select drectfast" is sent on mouse motion, and "select drect" is sent once when the mouse is released, so these callbacks are sent once on completion of a selection.

SVC:SHEET: n [f| b| s] m[+] The current sheet n, if it is front,back, or simplex, and the total sheet count m. + indicase page counting is not complete

The following SVC:STAMPxxx callbacks are generated when a file stamped by SwiftStamp is loaded with an "Idoc" command.

SVC:STAMPINFO:str The description of the stamping organization

SVC:STAMPDATE:str The date the file was stamped

SVC:STAMPLIC:str The ID of the SwiftStamp license

SVC:STAMPHOST:str The hostname where the file was stamped

SVC:UNITS inches| centimeters Current units per the units command

The following SVC:USERxxx callbacks are sent after the user completes an action initiated by the "gui userxxx start" commands.

SVC:USERCANCEL	User canceled input mode
SVC:USERDEF:x1 y1 x2 y2	After "gui userdef" is complete, the user-defined location in drawing coordinates (x1, y1) - (x2, y2)
SVC:USERDISPLAYDEF:x1 y1 x2 y2	After "gui userdef" is complete, the user-defined location in display coordinates (x1, y1) - (x2, y2)
SVC:USERDISPLAYSHOW:x1 y1 x2 y2	Current mouse position in display coordinates (x1,y1) - (x2,y2)
SVC:USERSHOW:x1 y1 x2 y2	Current mouse position in drawing coordinates (x1,y1) - (x2,y2)

These above two commands are sent after the "gui userdef" command is given, while the mouse is down, to allow the client to display appropriate feedback (using markup).

SVC:USERDISPLAYLINE:x0 y0 x1 y1	User defined line in display coordinates (x0,y0) - (x1,y1)
SVC:USERLINE:rx0 ry0 rx1 ry1	User defined line in drawing coordinates (rx0,ry0) - (rx1,ry1)

SVC:USERDISPLAYPOINT:x0 y0	User defined point in display coordinates (x0,y0)
SVC:USERPOINT:rx0 ry0	User defined point in drawing coordinates (rx0,ry0)
SVC:USERDISPLAYRECTANGLE:x0 y0 x1 y1	User defined rectangle in display coordinates (integer pixels) (x0,y0) - (x1,y1)
SVC:USERRECTANGLE:rx0 ry0 rx1 ry1	User defined rectangle in drawing coordinates (rx0,ry0) - (rx1,ry1)
SVC:VERSION: n	Return the SwiftView version number string
SVC:WINSIZE: xa ya max xp yp	New window size where: xa,ya = x,y aspect ratio – double max = 0.0 - 1.0 (percent of screen to use) xp,yp = window position on screen (-1.0 to +1.0) - optional
SVC:WORKING:ON OFF	Turn on/off "hourglass" cursor
SVC:WVIEW:width height xres yres	Current view windows size. Listed in integer pixels, and the display device x and y resolution (as real numbers) used to compute markup device coordinates, in the current units (in/cm per the "units" command) per pixel.

Appendix A: PCL Fonts

Overview

SwiftView can view HP Printer Control Language (PCL) files and convert those files to TIFF and other image formats. PCL is the language of HP LaserJet and compatible printers and PCL files can be generated by any MS Windows application and most UNIX applications. See the discussion above under "file type support" for additional PCL information.

Fonts are the only source of complexity in SwiftView's PCL support. PCL printers can supply or be supplied fonts in any or all of the following manners:

- Downloaded in the print stream from the application program or system
- Internal fonts - scaleable on newer printers
- Local font cartridges - including adding scaleable fonts on newer printers

SwiftView can also find fonts supplied at an internet location, provided the environment variables NDGTTFONT or NDGBMFFONT are defined on the client machine or in an ICS command sent to the client machine.

SwiftView does not have the internal font scaling engine provided with all HP printers starting with the LaserJet 3, but does have numerous font support capabilities, including download font support, access to TrueType fonts on Windows, and SwiftView-supplied font files on UNIX. For both UNIX and Windows, a minimum set of fonts is built into the program, including:

Courier 12 point in normal and bold in HP Roman8, PC-8, PC-8 D/N, and ISO 8859/1 Latin 1
Line printer 8.5 point normal in HP Roman8, PC-8, PC-8 D/N, and ISO 8859/1 Latin 1

Font selection

PCL printers select from available fonts according to the following priority:

1. Symbol set
2. Spacing (fixed or proportional)
3. Pitch (for fixed pitch only)
4. Height
5. Style (italic, normal, condensed, shadow, etc.)
6. Stroke weight (normal, bold, etc.)
7. Typeface family
8. Location (soft font highest)
9. Orientation (landscape and portrait)

Lower priority properties are considered only when choosing between fonts that exactly match the higher priority properties.

"Symbol set" is the HP technical name for the mapping of 8 bit (0-255) ASCII codes to the actual font character glyph. That is, it defines what shapes you see when the ASCII character value is printed or viewed. Most western symbol sets contain the same ASCII characters in the range 32-126 but provide considerable additional support for various symbols, drawing lines, and native language characters in the range 128-255.

Note that symbol set is the most significant factor in selecting a font and the shape of the typeface (Times Roman vs. Helvetica) is nearly the least important. This means that availability of or use of a given symbol set will have a large impact on the appearance of the page.

Symbol sets that SwiftView supports are:

- PC-8
- PC-850
- PC-8 Danish-Norwegian
- ISO 8859/1 Latin-1
- Windows 3.0 Latin-1
- Windows 3.1 Latin-1
- HP Roman-8
- Ventura-US
- Ventura-INTL
- Various ISO standard 7-bit European sets.

Any other symbol sets are defaulted to HP ROMAN-8, the default LaserJet font. External font files for HP Roman-8, HP PC-8 Danish-Norwegian, Latin-1, or PC-8 in specific point sizes are available on <http://www.swiftview.com>

PCL printers and SwiftView select the next smaller font if an exact match is not available.

On Windows:

When comparing the requested font properties to the Windows TrueType fonts, typefaces are mapped to Windows TrueType faces according to the following table. The first typeface listed is considered an exact match:

<u>PCL typeface/spacing</u>	<u>TrueType face used</u>
All fixed-space	Courier New
Courier (3) Lineprinter (0)	Courier New
Adobe Symbol (302)	Symbol
Wingdings (2730)	Wingdings
Arial (218), Univers (52)	Arial
Times Roman (5), all others	Times New Roman

PC-8 uses a few symbols from the built-in PC-8 Courier fonts for other typefaces because it is implemented by remapping the Windows 3.1 Latin-1 symbol set.

Roman-8 and PC-8 lack a few symbols because they are implemented by remapping Windows 3.1 Latin-1. A rubout glyph (a grid or an empty box) is substituted for any missing characters.

Typefaces mapped to Times New Roman do not support the Windows 3.1 and Windows 3.0 Latin-1 symbol sets, only Latin-1. Therefore for some Roman-8, Ventura, or PC-850 characters nothing will show up (not even a rubout glyph).

On UNIX:

All symbol sets other than Roman-8 and PC-8 are mapped to Latin-1. Roman-8 and PC-8 are complete because font files are provided. When a Latin-1 font file that better matches the requested properties is available, it is used for whatever characters it can provide, and the remaining characters come from the closest matching font file of the correct symbol set.

Ventura, PC-850, and Windows 3.x Latin-1 lack a few symbols because they are implemented by remapping Latin-1. A rubout glyph (a grid or an empty box) is shown for the missing characters.

Downloaded fonts

Placing the font definitions in the print stream or the resulting file guarantees that SwiftView will have all font information it requires. This is the recommended approach whenever possible. PCL files with downloaded fonts can be generated by any MS Windows program (using TrueType fonts) and many UNIX programs.

Windows TrueType

On MS Windows systems, SwiftView uses TrueType to generate bitmaps for fonts not downloaded or in font files. While these fonts may not be identical to those originally selected by the file, they will often be indistinguishable when viewed.

Font file overview

We supply font files with UNIX SwiftView only. No external font files are provided with the Windows product because it substitutes bitmaps generated from Windows resident TrueType files. These provide a complete set of LaserJet 3 internal fonts in discrete sizes and are not scaleable. All supplied PCL font files are supplied in 300 dpi only and located in the "p300x300" directory under the location of the SwiftView binary. On UNIX this is \$NDGUTIL/p300x300. This p300x300 directory can also be used to hold font files for Windows as well in cases where supplemental fonts will improve the appearance of a document. These font files can be downloaded from <http://www.swiftview.com/tech/supportfiles.htm>.

Scalable TrueType Font Support

In versions 5.3.0 and higher, SwiftView looks for TrueType font files in the directory \$NDGUTIL\ttf. TrueType fonts have the advantage of providing all point sizes in a single file as opposed to bitmap fonts. We provide additional TTF fonts for SwiftView use on our website here:

<http://www.swiftview.com/tech/supportfiles.htm>

TTF fonts can be used to replace fonts defined in a file, or for adding your own markup on a file. This setup requires a bit of customization, please contact tech@swiftview.com if you have questions.

Files in p300x300 directory

The file "fonts.txt" must be present containing the list of all font files (one per line of text) in the directory in any order. It can be generated on UNIX by typing "ls > fonts.txt", placing each name on a separate text line. Each font file contains the bitmaps for a single combination of the characteristics reflected in the name. Each name is of the form: TTWSSLXX.PPP

- TT Typeface:
 UN = Universe - LJ 3 internal font similar to HV
 TR = Times Roman or equivalent serif
 CR = Courier or equivalent monospace typewriter font
 LP = Line printer
 HV = Helvetica or equivalent sanserif
- W Style:
 R = Regular
 B = Bold
 I = Italic
- SS Symbol set:
 AU = HP PC-8
 8U = HP Roman-8
 ON = Latin 1 (ECMA-94)
 BU = HP PC-8 Danish/Norwegian
- L Orientation - only portrait fonts are required:
 L = Landscape
 P = Portrait

XX Numeric value indicates fixed pitch:

10 = 10 cpi

12 = 12 cpi

16 = 16.67 cpi

PS = Proportional spacing

PPP Point size. Examples are:

095 = 9.5 point

120 = 12 point

Examples:

CRR8UP10. 120 = Courier regular, Roman-8, 10 cpi, 12 point

UNB8UPPS. 180 = Universe bold, Roman-8, proportional spaced, 18 point

Font files supplied by SwiftView

The SwiftView executable program contains the LaserJet printer default fonts, enabling text files to be viewed without external font files. These fonts correspond to the font files:

crr8up10. 120 Courier regular, Roman-8, 10 cpi

crb8up10. 120 Courier bold, Roman-8, 10 cpi

crraup10. 120 Courier regular, PC-8, 10 cpi

crbaup10. 120 Courier bold, PC-8, 10 cpi

crr0np10. 120 Courier regular, Latin-1, 10 cpi

crb0np10. 120 Courier bold, Latin-1, 10 cpi

crrbup10. 120 Courier regular, PC-8 Danish/Norwegian, 10 cpi

crbbup10. 120 Courier bold, PC-8 Danish/Norwegian, 10 cpi

lpr8up16. 085 Line printer reg., Roman-8, 16.67 cpi

lpraup16. 085 Line printer reg., PC-8, 16.67 cpi

lpr0np16. 085 Line printer reg., Latin-1, 16.67 cpi

lprbup16. 085 Line printer reg., PC-8 Danish/Norwegian, 16.67 cpi

Font files supplied with the standard SwiftView distribution:

Proportional Universe (sanserif) regular style Roman-8

unr8supps. 040 unr8supps. 050 unr8supps. 060 unr8supps. 080

unr8supps. 100 unr8supps. 120 unr8supps. 140 unr8supps. 180

Proportional Univers (sanserif) bold style Roman-8

unb8supps. 040 unb8supps. 050 unb8supps. 060 unb8supps. 080

unb8supps. 100 unb8supps. 120 unb8supps. 140 unb8supps. 180

Proportional Univers (sanserif) italic style Roman-8

uni8supps. 080 uni8supps. 100 uni8supps. 120

Proportional CG Times (serif) regular style Roman-8

trr8supps. 040 trr8supps. 050 trr8supps. 060 trr8supps. 080

trr8supps. 100 trr8supps. 120 trr8supps. 140 trr8supps. 180

Proportional CG Times (serif) bold style Roman-8

trb8supps. 040 trb8supps. 050 trb8supps. 060 trb8supps. 080

trb8supps. 100 trb8supps. 120 trb8supps. 140 trb8supps. 180

Proportional CG Times (serif) italic style Roman-8

tri8supps. 080 tri8supps. 100 tri8supps. 120

Font files available at: <http://www.swiftview.com>

These would be used by European applications in particular for the Latin-1 symbol sets.

Proportional Univers (sanserif) regular style Latin-1

unr0npps. 040 unr0npps. 050 unr0npps. 060 unr0npps. 080
unr0npps. 100 unr0npps. 120 unr0npps. 140 unr0npps. 180

Proportional Univers (sanserif) bold style Latin-1

unb0npps. 040 unb0npps. 050 unb0npps. 060 unb0npps. 080
unb0npps. 100 unb0npps. 120 unb0npps. 140 unb0npps. 180

Proportional Univers (sanserif) italic style Latin-1

uni0npps. 080 uni0npps. 100 uni0npps. 120

Proportional CG Times (serif) regular style Latin-1

trr0npps. 040 trr0npps. 050 trr0npps. 060 trr0npps. 080
trr0npps. 100 trr0npps. 120 trr0npps. 140 trr0npps. 180

Proportional CG Times (serif) bold style Latin-1

trb0npps. 040 trb0npps. 050 trb0npps. 060 trb0npps. 080
trb0npps. 100 trb0npps. 120 trb0npps. 140 trb0npps. 180

Proportional CG Times (serif) italic style Latin-1

tri0npps. 080 tri0npps. 100 tri0npps. 120

Appendix B: Paper size to PCL ID map

The following is a map that specifies the Name, PCL ID, Height, and Width for standard paper sizes built into SwiftView.

Name	PCL ID	Width Inches/ Centimeters	Height Inches / Centimeters
Letter	2	8.5 / 216	11 / 280
Legal	3	8.5 / 216	14 / 356
A3	27	11.17 / 297	16.5 / 420
A4	26	8.3 / 210	11.7 / 297
A5	25	5.8 / 148	8.3 / 210
Executive	1	7.25 / 184	10.5 / 267
Ledger	6	17 / 432	11 / 279
Com10	81	9.5 / 24.13	4.12 / 10.46
C5	91	6.4 / 162	9 / 229
DL	90	8.66 / 22	4.33 / 11
Monarch	80	7.5 / 19.05	3.87 / 9.83
B5	100	6.9 / 176	9.8 / 250
Custom	101	11 / 27.94	7.5 / 19.05
JISB4	46	10.12 / 257	14.33 / 364
JISB5	45	7.17 / 182	10.12 / 257
JPOST	71	5.83 / 14.81	3.94 / 10
JPOSTD	72	7.87 / 20	3.83 / 9.73
Cheshire	910	11 / 279.4	14.9 / 378.46
Pressure1Up	911	3 / 76.2	6.5 / 165.1
Pressure4Up	912	12 / 304.8	14.9 / 378.46
Sackbag	913	7.5 / 190.5	14.9 / 378.46
170x125	920	17 / 431.8	12.5 / 325.12
136x84	921	13.6 / 345.44	8.4 / 213.36
17x43	922	17 / 431.8	4.3 / 109.22

Alphabetical Index of ICS Commands

Chapter 1: The Imaging Command Set.....	3
Manual Conventions.....	3
Product Specific Commands.....	4
Coordinate Systems.....	4
ICS parameter quoting.....	4
ICS Commands.....	5
<ad>.....	5
browse.....	5
callback filename [fn none].....	5
dpage [n first last next previous].....	5
draw.....	6
draw [all last].....	6
draw [lasttop lastbottom].....	6
draw [top bottom].....	6
draw [wide widetop].....	6
draw widebottom.....	6
get filename.....	6
get plotters.....	6
get version.....	6
gui autostart [enable disable].....	6
gui borderwidth n.....	6
gui button label ls icon in info tts command ics.....	7
gui clientmessage str.....	7
gui controls clear.....	7
gui color colorname rgb:red/green/blue.....	7
gui controls display [rightbuttons topbuttons none].....	8
gui controls reset.....	8
gui controls style [labels icons both].....	8
gui controls update.....	8
gui dialog activate.....	8
gui dialog name.....	8
gui dialog scrollmessage message str label str title str xpos n ypos n height n width n.....	9
gui dialog viewmode.....	10
gui displayduplex [enable disable].....	10
gui inline [enable disable].....	10
gui key keyname ics.....	10
gui key control keyname ics.....	10
gui logo [enable disable].....	11
gui logoicon [iconname default].....	11
gui mainwin ID.....	11
gui pagecontrol [enable disable].....	11
gui scrollbars action [page pan both].....	11
gui scrollbars display [vertical horizontal both none].....	12
gui scrollbars dynamic [enable pan disable].....	12
gui tooltips [enable disable].....	12
gui toolbarprops [enable disable].....	12
gui usercancel.....	12
gui userdef start.....	12
gui userline start.....	12
gui userpoint start.....	12

gui userrectangle start	12
gui winstate [hide maximize minimize restore]	13
gui wintitle str	13
hpgl [pen n all] red ccc green ccc blue ccc width w	13
hpgl startsoon [enable disable]	13
initdoc	13
ldoc fn	13
lpage [n first last next previous]	14
markup attributes coordinates c xorigin n yorigin n	14
markup attributes defaults	14
markup attributes display [yes no]	14
markup attributes drawmode xor	14
markup attributes linestyle ls fillpattern fp fillshade fs	15
markup attributes replaceonly colorname	15
markup attributes dontreplace colorname	15
markup attributes dontdraw colorname	15
markup attributes drawonly colorname	15
markup attributes drawmode opaque	15
markup attributes drawmode transparent	15
markup attributes bgcolor colorname	16
markup attributes fgcolor colorname	16
markup attributes background colorname	16
markup attributes foreground colorname	16
markup attributes print [yes no]	16
markup attributes background push	16
markup attributes foreground pop	16
markup entity hotspot ics status str	16
markup circle rxloc x ryloc y radius r	17
markup filledcircle rxloc x ryloc y radius r	17
markup clear transient	17
markup clear all	17
markup clear last	17
markup crop left lmargin right rmargin top tmargin bottom bmargin	17
markup crop rrect x0 y0 x1 y1	17
markup display markups	17
markup display original	17
markup display both	17
markup line x0 y0 ... xn yn	18
markup arrow x0 y0 ... xn yn	18
markup load filename fn rxloc x ryloc y scale s rotate r	18
markup pageinit	18
markup polygon x0 y0 ... xn yn	18
markup filledpolygon x0 y0 ... xn yn	18
markup rectangle x0 y0 x1 y1	18
markup filledrectangle x0 y0 x1 y1	18
markup reset	18
markup text font "... " rxloc x ryloc y string "... "	18
markup text rotate n	19
markup units [inches centimeters]	19
onpage ppp ics	19
onpage ppp use sourcename fn lpage [enable disable]	20
onpage ppp use blankpage size lpage [enable disable]	20
onpage [odd even front back range 1-n simplex duplex long short]	20
onpage reset	20
pagesize rxsize xs rysize ys	20
pagesize original	20

pan [up down left right]	20
pan [x p y p]	20
pcl defaultres [300 600]	21
pcl defaultduplex [long short original]	21
pcl duplex [enable disable]	21
pcl edge-to-edge [on off]	21
pcl forceres [0 300 600 1200]	21
pcl initfile fn	21
pcl linefeed [cr_lf lf]	21
pcl [letter legal a4 a3]_out_tray [n default]	22
pcl oldfonts off	22
pcl pagesize n orientation o [linesperpage fontpitch] p reset	22
pcl passduplex [enable disable]	23
pcl passtray [enable disable]	23
pcl passpapersize [enable disable]	23
plot pn [all m-n current rangefile:fname] [cut x1 y1 x2 y2 rcut x1 y1 x2 y2 view odd even reverse] [onefile]	23
printer number pn type pt command cs alias as	23
printer default pn	24
quit	24
reload	24
resetcache	25
resetpages	25
rlinstall [product {productname}] file {file/url} type {type_num}	25
rotate [n next]	25
save ft [all m-n current rangefile:fname] fn [dcut x1 y1 x2 y2 rcut x1 y1 x2 y2 view odd even reverse onefile]	25
save textpos option fn	26
save charpos option fn	26
scale all	26
scroll [up down left right] [n default]	27
select cancel	27
select copy	27
select [rrect drect] x0 y0 x1 y1 rect [enable disable]	27
search [rrect drect] x0 y0 x1 y1 rect [enable disable]	27
select string str matchcase [enable disable] regexp [enable disable] from [firstpage currentpage currentpageonly] found foundifcurrent]	27
search string str matchcase [enable disable] regexp [enable disable] from [firstpage currentpage currentpageonly] found foundifcurrent]	27
select next	28
search next	28
set allpensblack [enable disable]	28
set automarkup onpage [enable disable]	28
set [bitonalgamma dgamma] absval g	28
set [bitonalgamma dgamma] relval v	29
set [bitonalgamma dgamma] relrange r	29
set bitonalscaling [bestquality bestspeed false]	29
set browser service	29
set cachebrowser [on off]	29
set cachepcl [readonly readwrite memoryonly none]	29
set cachesize [NNNN default]	29
set cacheblocksize [NNNN default]	30
set cachereset [enable disable]	30
set checktext [enable disable]	30
set colorpalette [nochange sysdefault]	30
set count001pages [enable disable pattern]	30

set dialogs [enable disable]	31
set displayautorotate [enable disable].....	31
set docinfo [Title Author Subject Keywords Creator Producer CreationDate ModDate]	31
set docname fn	31
set dps [enable disable]	31
set displaycolor [color grey gray bitonal enable disable]	31
set env str	31
set filename fn.....	31
set filetype [pcl hpgl tiff unixtext dostext pdf default]	32
set fractionallinefeed pct	32
set fractionalspace n.....	32
set invertimage [true false].....	32
set jbig2_maxcomp n [10-20000]	32
set jbig2_weight_factor d[0.0-1.0].....	32
set [ldocinit dpageinit draw drawtop drawbottom "..."]	32
set maxjbigobjects [1-1000000].....	33
set nofileerror [enable disable]	33
set onexit <cmd>.....	33
set output [enable disable].....	33
set pagespertick [1 default]	33
set pdfconfig <filename>	33
set pdfhintflags [enable disable].....	33
set pdfres n [60-1200]	34
set printarea [paper printable]	34
set printautorotate [enable disable]	34
set printcolor [color grey gray bitonal enable disable]	34
set printcompressformat [beststd std unc g3 g4 faxlo faxhi packbits jbig2 jpeg flate]	34
set printcompresslevel N.....	35
set printduplex [disable long short original]	35
set printinray[N original disable]	35
set printinraymap originaltray N tray N	35
set printinraymap originalsizename name tray N.....	36
set printinraymap originalsizepcl N tray N	36
set printinraymap originalsizename clear	36
set printinraywinmap pclid windriverid.....	36
set printjobname [filename default]	36
set printmenudefault method [fast direct] [enable disable]	36
set printmenu printtofile [off on].....	36
set printmenudefault printertype [pclps notp] passpapersize [y n] scale [1 all printable] color [enable disable] res [original n] duplex [long original disable]	36
set printouttray [N original off]	37
set printpassword s	37
set printpj [enable disable]	37
set printprerotate [enable disable]	38
set printres [n original]	38
set printscale [auto fit n]	38
set printsize [pcl n name s width x height y source original source windriver] xmargin x ymargin y orientation o oversize n undersize m.....	39
set printspotcolor [enable disable].....	39
set printtext [enable disable]	39
set printtiff fill n	39
set printwintray pclid winid	39
set printxposition [center left right]	40
set printyposition [center top bottom]	40
set [printxoffset printyoffset] n	40
set product {product_name}	40

set ps2output [recompress default]	40
set quotechar 0xNN	40
set rlminit {file_path}	41
set roam [on off]	41
set savetextlpi n	41
set select charset [unknown EBCDIC]	41
set select [drawing document]	41
set select [enable disable iftext]	41
set sie [enable disable]	42
set swiftservetimeout [t default]	42
set sourceimagephoto [missing override] v	42
set sourceimages [missing override] n	42
set textencoding [latin1 utf8]	42
set timedcommand repeat n ics	42
set zippassword str	42
set zoomfactor n	43
system s	43
units [inches centimeters]	43
use reset	43
winsize xa ya ws xp yp	43
zoom [in out]	43
zoom [rrect drect] x0 y0 x1 y1	43
zoom selectrect	43
Chapter 2: SwiftView Environment Variables	44
NDGCONFIG	44
NDGTTFONT	44
NDGBMFONT	44
NDGICS0, NDGICS1, NDGICS2,	44
NDGHPGL	44
NDGMACRO	44
NDGNOCUSTOMSIZE	44
NDGNOFILE	45
NDGNOSIE	45
NDGNOCOLOR	45
NDGONEPRINTJOB	45
NDGDEBUG	45
NDGDEBUG_FILE	45
NDG_PAS	45
NOR5I18N	45
SV_DISPLAY_RATIO	46
Chapter 3: ICS Callbacks	47
SVC:BASELINE:ryorigin	47
VC:COPYTEXT:str	47
SVC:COVER:TRUE FALSE	47
SVC:DIALOGS:ENABLE DISABLE	47
SVC:DOC:fn	47
SVC:DOCINFO:filetype width height fn	47
SVC:DUPLEX: [long short simplex]	48
SVC:DVVIEW:dx0 dy0 dx1 dy1 vx0 vy0 vx1 vy1 rescale	48
SVC:ERROR:str	48
SVC:FILENAME:fn	48
SVC:GUI:...command tokens...	48
SVC:GUI:clientmessage str	48
SVC:IMPRESSION:0 m[+]	48
SVC:MESSAGE:str	48
SVC:NOTICE:str	48

SVC:OUTPUT:CURRENT:cp COUNT np LIST: printernum alias	48
SVC:OUTPUT:START:str	49
SVC:OUTPUT:STOP	49
SVC:PAGE:cp np	49
SVC:PAGEDOC: filename	49
SVC:PAGEINFO: resolution n [bitonal gray color] compression [G4 JBIG2 JPEG FLATE]	49
SVC:QUIT	49
SVC:RESIZE	49
SVC:ROTATE: n	49
SVC:RVIEW:xul yul xlr ylr	49
SVC:SELECT:SCROLL:panval SVC:SEARCH:SCROLL:panval	50
SVC:SELECT:LOCATION:pnum rx1 ry1 rx2 ry2 str	50
SVC:SEARCH:LOCATION:pnum rx1 ry1 rx2 ry2 str	50
SVC:SELECT:STRING:str	50
SVC:SEARCH:STRING:str	50
SVC:SHEET: n [f b s] m[+]	50
SVC:STAMPINFO:str	50
SVC:STAMPDATE:str	50
SVC:STAMPLIC:str	50
SVC:STAMPHOST:str	50
SVC:UNITS inches centimeters	50
SVC:USERCANCEL	51
SVC:USERDEF:x1 y1 x2 y2	51
SVC:USERDISPLAYDEF:x1 y1 x2 y2	51
SVC:USERDISPLAYSHOW:x1 y1 x2 y2	51
SVC:USERSHOW:x1 y1 x2 y2	51
SVC:USERDISPLAYLINE:x0 y0 x1 y1	51
SVC:USERLINE:rx0 ry0 rx1 ry1	51
SVC:USERDISPLAYPOINT:x0 y0	52
SVC:USERPOINT:rx0 ry0	52
SVC:USERDISPLAYRECTANGLE:x0 y0 x1 y1	52
SVC:USERRECTANGLE:rx0 ry0 rx1 ry1	52
SVC:VERSION: n	52
SVC:WINSIZE: xa ya max xp yp	52
SVC:WORKING:ON OFF	52
SVC:WVIEW:width height xres yres	52
Appendix A: PCL Fonts	53
Overview	53
Font selection	53
Downloaded fonts	54
Windows TrueType	55
Font file overview	55
Scalable TrueType Font Support	55
Files in p300x300 directory	55
Font files supplied by SwiftView	56
Font files supplied with the standard SwiftView distribution:	56
Font files available at: http://www.swiftview.com	57
Appendix B: Paper size to PCL ID map	58