

SunPHIGS 3.0

Sun-specific Reference Manual



Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043

Part No: 801-4367-10
Revision A, April 1993
SunPHIGS Release 3.0

© 1993 Sun Microsystems, Inc. – Printed in the United States of America.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc. and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's Font Suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, SMCC, the SMCC logo, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. SunPHIGS, SunPEX, and XGL are trademarks of Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK ® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark and product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

XGL, XView are trademarks of Sun Microsystems, Incorporated.

NAME	INTRODUCTION TO SUNPHIGS TABLES AND EXPLANATIONS – the introduction to tables and explanations in the SunPHIGS graphics library
DESCRIPTION Tables and Explanations	<p>The following tables and explanations are provided, accompanied by the appropriate cross-reference name:</p> <p>CGM cgm (7P) COLOUR colour (7P) INTRO DRAWABLE REGION intro_drawable_region (7P) INTRO INTERNATIONALIZATION intro_internationalization (7P) PHIGS DESCRIPTION TABLE phigs_description_table (7P) PHIGS TRAVERSAL STATE LIST phigs_traversal_state_list (7P) PHIGS WORKSTATION DESCRIPTION TABLE phigs_workstation_description_table (7P)</p> <p>CGM – contains information describing mapping of SunPHIGS functions to CGM elements.</p> <p>COLOUR – introduction to using colour in SunPHIGS graphics library</p> <p>INTRO DRAWABLE REGION – the introduction to the <i>X Drawable</i> region workstation type</p> <p>INTRO INTERNATIONALIZATION – introduction to the <i>internationalization</i> extensions to the SunPHIGS graphics library</p> <p>PHIGS DESCRIPTION TABLE – contains information describing the static capabilities of SunPHIGS</p> <p>PHIGS TRAVERSAL STATE LIST – contains all transient information SunPHIGS uses during traversal</p> <p>PHIGS WORKSTATION DESCRIPTION TABLE – contains information describing the static capabilities of a SunPHIGS workstation</p>
SEE ALSO	<p>CGM (7P) COLOUR (7P) INTRO DRAWABLE REGION (7P) INTRO INTERNATIONALIZATION (7P) PHIGS DESCRIPTION TABLE (7P) PHIGS TRAVERSAL STATE LIST (7P) PHIGS WORKSTATION DESCRIPTION TABLE (7P) INTRO (3P)</p>

INTRO (3PP)
INTRO PHIGS (3P)

NAME	CGM – contains information describing mapping of SunPHIGS functions to CGM elements.
DESCRIPTION Overview	The Computer Graphics Metafile (CGM) standard defines a metafile for the capture of static picture definitions. Since use of the CGM was not intended to be restricted to PHIGS environments, a one-to-one mapping between the two standards is not possible in all cases. This document describes the generation of CGM in a SunPHIGS environment.
Application Profiles	Application profile (AP) defines a subset of permissible values for parameters of elements to ease interoperability of implementations of a standard. The Technical Office Protocol (TOP) CGM Application Profile defines the use of binary and long-format command headers and restricts the use of private ESCAPES and GDPs. SunPHIGS generates TOP conforming metafiles.
Encodings	<p>Three standard encodings defined in CGM are:</p> <ul style="list-style-type: none"> • Character Encoding • Binary Encoding • Clear Text Encoding <p><i>Character Encoding</i> provides minimum size encoding. It is designed for transfer through networks that cannot support binary transfers. <i>Binary Encoding</i> provides an encoding requiring the least effort to generate and interpret on many systems. <i>Clear Text Encoding</i> provides an encoding that can be created, viewed, and edited with standard text editors. It is, therefore, also suitable for transfer through networks that support only transfer of text files. SunPHIGS can generate any of these encodings. See WORKSTATION TYPE SET (3P).</p>
Picture Generation	CGM is generated by a workstation of type MO. See OPEN WORKSTATION (3P). At appropriate times, the PHIGS structures posted on an MO/CGM workstation are traversed, and CGM elements are created. During PHIGS traversal, PHIGS primitives, which are in three dimensional space, are clipped to the view volume and then projected onto the view plane. Clipped PHIGS primitives may result in the generation of one or multiple CGM elements. Workstation transformation is applied to the clipped-and-projected primitives, and appropriate CGM elements are generated.
CGM Construct	<p>A CGM created through SunPHIGS is usually in the following construct:</p> <ul style="list-style-type: none"> • BEGIN METAFILE • Metafile Descriptor Elements • BEGIN PICTURE • Attribute Elements • Graphical Primitive Elements • END PICTURE • END METAFILE

Metafile Descriptor Elements

Metafile Descriptor elements serves a similar purpose as the Workstation Description Table in PHIGS. By default, the following *Metafile Descriptor* elements appear at the beginning of every CGM metafile generated by SunPHIGS:

<i>Metafile Elements</i>	<i>Default Value</i>
METAFILE VERSION	1
METAFILE DESCRIPTION	Sun Microsystems CGM, TOP/FULL conformance
METAFILE ELEMENT LIST	list of metafile elements used in the metafile
VDC TYPE	INTEGER
INTEGER PRECISION	-32767 32767
REAL PRECISION	-32767.0 32767.0 4
INDEX PRECISION	0 127
COLOUR PRECISION	255
COLOUR INDEX PRECISION	255
COLOUR VALUE EXTENT	0 0 0 255 255 255
MAXIMUM COLOUR INDEX	255
METAFILE DEFAULTS REPLACEMENT	list of metafile defaults

Attribute Elements

Setting of the POLYLINE, POLYMARKER, TEXT, INTERIOR and EDGE representations on MO/CGM workstations is recorded in the MO workstation state list. Because these representations cannot be encoded in a CGM metafile, PHIGS output primitives using bundled values are encoded in the metafile using individual attribute control. The following table shows the mapping of the SunPHIGS attribute control functions to CGM attribute elements:

<i>SunPHIGS Attribute Functions</i>	<i>CGM Elements</i>
SET LINETYPE	LINE TYPE
SET LINE WIDTH SCALE FACTOR	LINE WIDTH
SET POLYLINE COLOUR INDEX	LINE COLOUR
SET MARKER TYPE	MARKER TYPE
SET MARKER SIZE SCALE FACTOR	MARKER SIZE
SET POLYMARKER COLOUR INDEX	MARKER COLOUR
SET INTERIOR STYLE	INTERIOR STYLE
SET INTERIOR STYLE INDEX	PATTERN INDEX or HATCH INDEX
SET INTERIOR COLOUR INDEX	FILL COLOUR

Graphical Primitive Elements

The following table shows mapping of the SunPHIGS output primitives to CGM Graphical Primitive elements:

<i>SunPHIGS primitives</i>	<i>CGM elements</i>
POLYLINE, POLYLINE3, POLYMARKER, POLYMARKER 3, TEXT, TEXT 3, ANNOTATION TEXT RELATIVE, ANNOTATION TEXT RELATIVE 3	One or multiple POLYLINE elements
FILL AREA, FILL AREA 3	POLYGON
FILL AREA SET, FILL AREA SET 3	POLYGON SET, and POLYLINE if edge flag is ON
CELL ARRAY, CELL ARRAY 3,	One or multiple POLYLINE elements
GDP - CIRCLE, GDP - CIRCLE 3	POLYGON SET, and POLYLINE if edge flag is ON
GDP - ANNOTATION CIRCLE, GDP - ANNOTATION CIRCLE 3	CIRCLE, and POLYLINE if edge flag is ON
GDP - CIRCULAR ARC, GDP - CIRCULAR ARC 3	One or multiple POLYLINE elements
GDP - ANNOTATION CIRCULAR ARC, GDP - ANNOTATION CIRCULAR ARC 3	CIRCULAR ARC CENTRE
GDP - CIRCULAR ARC CLOSE, GDP - CIRCULAR ARC CLOSE 3	POLYGON SET, and POLYLINE if edge flag is ON
GDP - ANNOTATION CIRCULAR ARC CLOSE, GDP - ANNOTATION CIRCULAR ARC CLOSE 3	CIRCULAR ARC CENTRE CLOSE
GDP - ELLIPSE, GDP - ELLIPSE 3	POLYGON SET, and POLYLINE if edge flag is ON
GDP - ANNOTATION ELLIPSE, GDP - ANNOTATION ELLIPSE 3	ELLIPSE, and POLYLINE if edge flag is ON
GDP - ELLIPTICAL ARC, GDP - ELLIPTICAL ARC 3	One or multiple POLYLINE elements

This table is continued on the next page

<i>SunPHIGS primitives</i>	<i>CGM elements</i>
GDP - ANNOTATION ELLIPTICAL ARC, GDP - ANNOTATION ELLIPTICAL ARC 3	ELLIPTICAL ARC
GDP - ELLIPTICAL ARC CLOSE, GDP - ELLIPTICAL ARC CLOSE 3	POLYGON SET, and POLYLINE if edge flag is ON
GDP - ANNOTATION ELLIPTICAL ARC CLOSE, GDP - ANNOTATION ELLIPTICAL ARC CLOSE 3	ELLIPTICAL ARC CLOSE, and POLYLINE if edge flag is ON
GDP - RECTANGULAR GRID, GDP - RECTANGULAR GRID 3, GDP - RADIAL GRID, GDP - RADIAL GRID 3	One or multiple POLYLINE elements
GDP - MONO-ENCODED TEXT, GDP - MONO-ENCODED TEXT 3, GDP - MONO-ENCODED ANNOTATION TEXT RELATIVE, GDP - MONO-ENCODED ANNOTATION TEXT RELATIVE 3	One or multiple POLYLINE elements
POLYLINE SET 3 WITH DATA †	One or multiple POLYLINE elements
NON-UNIFORM B-SPLINE CURVE †	POLYMARKER or POLYLINE
FILL AREA SET 3 WITH DATA, † TRIANGLE STRIP 3 WITH DATA, † QUADRILATERAL MESH 3 WITH DATA, † SET OF FILL AREA SET 3 WITH DATA, †	POLYGON SET, and POLYLINE if edge flag is ON. Vertex colours, vertex normals, facet colours, facet normals, and PHIGS PLUS rendering attributes (such as lighting, shading, and depth cueing) are ignored.
NON-UNIFORM B-SPLINE SURFACE †	

† These are *SunPHIGS Extensions* based on PHIGS PLUS and are not part of the PHIGS standard.

Line Types mapping

Of the eight line types defined in SunPHIGS, only five have equivalents defined in CGM. Line types that are not defined in CGM are mapped to the line types that most closely resemble them. The following table shows mapping of SunPHIGS line types to CGM line types:

<i>SunPHIGS Line Types</i>		<i>CGM Line Types</i>	
Value	Meaning	Value	Meaning
1	Solid	1	Solid
2	Dashed	2	Dashed
3	Dotted	3	Dotted
4	Dash-dotted	4	Dash-dotted
0	Long-dashed	2	Dashed
-1	Dot-dash-dotted	5	Dash-dot-dotted
-2	Center	5	Dash-dot-dotted
-3	Phantom	5	Dash-dot-dotted

Hatch Styles Mapping

Of the 24 hatch styles defined in SunPHIGS, only six have equivalents defined in CGM. Hatch styles that are not defined in CGM are mapped to the hatch styles that most closely resemble them. The following table shows mapping of SunPHIGS hatch styles to CGM hatch styles:

<i>SunPHIGS Hatch Styles</i>		<i>CGM Hatch Styles</i>	
Value	Meaning	Value	Meaning
-1	Horizontal	1	Horizontal
-2	Vertical	2	Vertical
-3	45 Degree-angled	3	45 Degree-angled
-4	135 Degree-angled	4	135 Degree-angled
-5	Rectangular Grid	5	Rectangular Grid
-6	Diagonal Grid	6	Diagonal Grid
-51	Double-spaced Horizontal	1	Horizontal
-52	Double-spaced vertical	2	Vertical
-53	Double-spaced 45 Degree-angled	3	45 Degree-angled
-54	Double-spaced 135 Degree-angled	4	135 Degree-angled
-55	Double-spaced Rectangular Grid	5	Rectangular Grid
-106	Double-spaced Diagonal Grid	6	Diagonal Grid
-101	Double-spaced Transparent Horizontal	1	Horizontal
-102	Double-spaced Transparent Vertical	2	Vertical
-103	Double-spaced Transp. 45 Degree-angled	3	45 Degree-angled
-104	Double-spaced Transp. 135 Degree-angled	4	135 Degree-angled
-105	Double-spaced Transp. Rectangular Grid	5	Rectangular Grid
-106	Double-spaced Transp. Diagonal Grid	6	Diagonal Grid

LIMITATIONS

CGM does not have facilities to define the image placement on a physical display surface; hence, the workstation viewport is discarded. Hidden line and hidden surface removal (HLHSR) is not supported in the MO/CGM workstation.

SEE ALSO

OPEN WORKSTATION (3P)
WORKSTATION TYPE SET (3P)
GET ITEM TYPE FROM METAFILE (3P)
READ ITEM FROM METAFILE (3P)
INTERPRET ITEM (3P)

NAME	COLOUR – introduction to using colour in SunPHIGS graphics library
DESCRIPTION	<p>This page provides an overview of using the capabilities of a colour system with SunPHIGS. The topics covered are:</p> <ul style="list-style-type: none"> • Colour Frame Buffers • Standard PHIGS: Indexed Colour • PHIGS PLUS: General Colour • A Workstation's Default Colour Table • Setting a Workstation's Colour Table • Sizing a Workstation's Colour Table • Colormap Double Buffering • Sharing Colormap Resources • Other Colour Attributes <p>Note: <i>colormap</i> is a general graphics term, rather than a PHIGS term, so it has the general spelling. Other references to color are PHIGS terms, so they have the British spelling <i>colour</i>.</p>
Colour Frame Buffers	<p>SunPHIGS supports several different colour frame buffers which have different colour capabilities. All of Sun's current frame buffers display pixels by specifying the red, green, and blue (RGB) components of the colour for each pixel. This specification can be done in two different ways, <i>indexed</i> and <i>true colour</i>.</p> <p>Indexed frame buffers have a hardware colour table, also known as a colormap, which holds colour definitions. Each pixel in the frame buffer holds a colour index which selects a colour definition from this colormap. The pixels are converted from indices to RGB colours each time the monitor is refreshed from the frame buffer (many times a second). The colour table holds 24-bit RGB values, so that each entry chooses a colour from a 16 million colour palette. This is close to the limit of human perception. Each pixel holds an 8-bit index into the colormap, so that each pixel can select one of the 256 entries in the colormap. This means that there can be a maximum of 256 different colours on an indexed frame buffer at a time, and that each of those colours can be chosen from a 16 million colour palette. Since the conversion from index to RGB colour is done each time the monitor is refreshed, changes to the colormap appear almost instantaneously. For more information, see the <i>Sizing a Workstation's Colour Table</i> section, below.</p> <p>Sun's true colour frame buffers store 24-bit RGB values for each pixel. This means that each pixel can take any of 16 million colours. Since each pixel specifies a RGB colour, there is no need for a colormap.</p> <p>SunPHIGS uses the colour mode workstation configuration attribute to indicate whether a workstation is open on an indexed or true colour frame buffer. The attribute is called PHIGS_COLOUR_MODE in C (with values PHIGS_INDEX_COLOUR and PHIGS_TRUE_COLOUR) and PHIGSCLRMODE in FORTRAN (with values PHIGSINDEXCOLOUR and PHIGSTRUECOLOUR). The default value is true colour. If the colour mode is indexed, then the workstation is called</p>

an indexed colour workstation, if it is true colour, the workstation is called a true colour workstation. SunPHIGS can simulate a true colour frame buffer on an indexed frame buffer; this is enabled by setting the colour mode to true colour on an indexed frame buffer. This simulation may be slower than a normal indexed or true colour workstation, and the appearance of primitives may be different. See WORKSTATION TYPE SET (3P).

**Standard PHIGS:
Indexed Colour**

Standard PHIGS supports only *indexed colour*. Structure elements select a colour for a primitive by using an integer colour index, such as that selected by SET POLYLINE COLOUR INDEX(3P). When a primitive is displayed on a workstation, the colour index is used to select a colour value (called a colour representation) from the workstation's colour table. The colour representation is typically three floating-point numbers, interpreted in a certain *colour model*, such as RGB (red, green, and blue components). The combination of these colour components is what is actually seen by the observer. The default colour index for all primitives is 1.

**PHIGS PLUS:
General Colour**

SunPHIGS extensions based on PHIGS PLUS allow the colour specification to be either indexed or RGB. When a colour is specified as an index, it is called *indirect colour*, since the colours are indirectly specified through the workstation colour table. When a colour is specified as an RGB triple, it is called *direct colour*. *General colour* is an abstraction that allows direct or indirect colour to be specified as a single aspect. A general colour consists of a type and a value. If the colour type is indirect, then the value is interpreted as an index into the workstation colour table. Otherwise, the color type indicates a color model, such as RGB. If the type is RGB, then the colour value is interpreted as an RGB value.

Using SunPHIGS extension primitives and attributes, the colour for a primitive can be set using either attribute elements, such as SET POLYLINE COLOUR (3PP), or primitives with one or more colours as part of the primitive, such as POLYLINE SET 3 WITH DATA (3PP). In either case, the colour is specified as a general colour, so that an index or direct colour can be used. This functionality supersedes the standard PHIGS functionality. Standard PHIGS primitives will be rendered using the PHIGS PLUS attributes, and standard PHIGS attributes affect PHIGS PLUS primitives. For example, SET POLYLINE COLOUR INDEX(3P) is equivalent to calling SET POLYLINE COLOUR(3PP) with the type set to indirect, both elements set the same attribute. The default colour for all primitives is indirect colour, index 1. If a primitive or attribute specifies direct colour on an indexed colour workstation, then *indirect colour, index 1* is used instead.

**A Workstation's
Default Colour Table**

The size of the colour table is workstation-dependent, but entries 0 and 1 are always defined. SunPHIGS predefines the colour table for all workstations to be:

<i>Index</i>	<i>Colour</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
0	black	0.0	0.0	0.0
1	white	1.0	1.0	1.0
2	red	1.0	0.0	0.0
3	green	0.0	1.0	0.0
4	blue	0.0	0.0	1.0
5	yellow	1.0	1.0	0.0
6	cyan	0.0	1.0	1.0
7	magenta	1.0	0.0	1.0
≥8	white	1.0	1.0	1.0

The colour at index 0 is the background colour. PHIGS clears the display surface to this colour index as the first step in regenerating an image.

Setting a Workstation's Colour Table

In PHIGS, an application can use the function SET COLOUR REPRESENTATION to specify a colour representation by a triple of real values whose meaning depends on the current colour model. SunPHIGS supports only the RGB colour model. The colour coordinate space is [0,1] in all three dimensions. No intensity is 0.0; maximum intensity is 1.0. The RGB values for the default colour definitions are given in the table above.

A true colour workstation has a workstation colour table and supports indirect colour.

A change to a colour representation shows up immediately on an indexed colour workstation and during the next regeneration on a true colour workstation.

A PHIGS application can inquire for any colour index, either the value it sets in the workstation's colour table or the value that will be *Realized* on the display, after mapping to available colour values. See INQUIRE COLOUR REPRESENTATION (3P).

Other colour models PHIGS defines are CIE (*Commission Internationale de l'Eclairage's* universal colour definition system), HSV (Hue, Saturation, and Value), and HLS (Hue, Lightness, and Saturation).

Colour representations 0 and 1 can be set on a monochrome workstation to produce either normal or inverse video.

Sizing a Workstation's Colour Table

In PHIGS, a workstation's colour table has a fixed size. SunPHIGS allows the size to be selected from any power of 2, up to and including 256 on indexed colour workstations or 4096 on true colour workstations. The default is 32 entries. Before opening the workstation, an application may select the size by using the workstation attribute PHIGS_COLOUR_TABLE_SIZE in C or PHIGSCLRTBLSIZE in FORTRAN. See WORKSTATION TYPE SET(3P).

On indexed colour workstations, the window system arbitrates the use of the hardware colormap among windows, by allocating to each window a logical colormap (or *virtual colormap*).

Note: The hardware colormap has a fixed size and is a limited resource that must be shared by all of the clients using a server.

When a virtual colormap is installed, any colormap that is currently installed in the hardware will be uninstalled, and the new colormap will be installed into the hardware. When the new colormap is installed, all of the clients who used the old colormap will be displayed in false colors.

It is the window manager's responsibility to install and uninstall colormaps according to its colormap focus policy.

Colormap Double Buffering

Double buffering is a simple concept. First, the complete picture is prepared; then, the complete picture is displayed. This produces a smoother switch from one frame to the next. The viewer does not see the drawing of the new frame. When a double buffered workstation traverses the posted structures, they are rendered into the undisplayed buffer. When the traversal is done and the image in the undisplayed buffer is complete, then the undisplayed buffer and displayed buffer are swapped. An application may specify whether quick update methods should be double buffered by using `ESCAPE -22 (3P)` and `ESCAPE -23 (3P)`. LP SunPHIGS applications control double buffering in C by using the workstation attribute `PHIGS_X_BUF_MODE` (values are `PHIGS_BUF_SINGLE`, `PHIGS_BUF_DOUBLE_CMAP`, and `PHIGS_BUF_DOUBLE`) and in FORTRAN by using `PHIGSDBLBUFFER` (values are `PHIGSBUFSINGLE`, `PHIGSBUFDBLCMAP`, and `PHIGSBUFDDOUBLE`). These values request the following alternatives: no double buffering, colormap double buffering, and hardware double buffering. No double buffering is the default. See `WORKSTATION TYPE SET (3P)`.

Colormap double buffering uses the colormap and the frame buffer write-enable mask to handle a single frame buffer as two virtual buffers. The only implementation detail that a SunPHIGS application needs to know is the following: the actual number of color calls that the window requires is equal to the square of the number of entries in the PHIGS colour table. (The size of the PHIGS colour table is the number of entries in one of the virtual buffers.)

A SunPHIGS workstation with an 8-entry colour table (3 bits) with colormap double buffering actually uses 64 colours (6 bits) in its colormap. A 16-entry (4-bit) colour table with colormap double buffering uses 256 colours (all 8 bits) of the frame buffer. The colormap is swapped in and out as the cursor moves in and out of the window. A SunPHIGS workstation with a 32-entry colour table (the default) cannot use colormap double buffering. No window sharing a colormap (see below) can use colormap double buffering. True colour workstations do not support colormap double buffering.

Hardware double buffering is an alternative technique that does not require extra colormap resources. It uses two physical frame buffers, only one of which is displayed.

Sharing Colormap Resources

The hardware colormap is a limited resource. Contention for this resource may cause colormap flashing. Thus, sharing of this resource among workstations can be extremely useful. If colour values are to be loaded once and never changed, and another window can use the same colour table definitions, sharing the colormap is expedient. However, colour tables used for animation, or for which colormap double buffering is enabled, should not be shared since the colours will change in all workstations sharing the

colormap.

Any application, whether it uses SunPHIGS or not, that assigns a different colormap to a subwindow of a top-level window needs to ensure that the subwindow's colormap is loaded when the mouse is moved into it. Rather than loading the subwindow's colormap on its own, the application should instead follow the conventions described in the *Inter-Client Communication Conventions Manual (ICCCM)* for informing the window manager about that subwindow's different color map. Section 6.4 of version 1.0 of this document states the following convention:

If a client has a top-level window that has sub-windows or override-redirect popup windows whose colormap requirements differ from the top-level window, that client should set a WM_COLORMAP_WINDOWS property on the top-level window. The WM_COLORMAP_WINDOWS property contains a list of the window IDs of windows that the window manager should track for colormap changes.†

† from *Inter-Client Communications Conventions Manual, Version 1.0, MIT X Consortium Standard*, by David S. H. Rosenthal, Sun Microsystems, Inc.

An ICCCM-compliant window manager will use the information provided in the WM_COLORMAP_WINDOWS property to load a subwindow's colormap when the mouse cursor is moved into that window; thus, the application need not worry about tracking those mouse entry events. Several SunPHIGS example programs, such as `cmap_panel.c`, illustrate the use of the WM_COLORMAP_WINDOWS property for informing the window manager about a subwindow's color map.

Sharing of colormap resources is performed very differently for indexed and true colour workstations.

True Colour Workstations

On a true colour workstation (the default) the colormap resource refers to the color cube used to simulate true colour on a PseudoColor visual. The workstation type attribute PHIGS_X_CMAP_PROP_ATOM gives the application access to a standard colormap stored as a property of the root window. This standard colormap is defined in `$OPENWINHOME/include/X11/xutil.h` as follows:

```

/*
 * This defines a window manager property that clients
 * may use to share standard color maps of type RGB_COLOR_MAP:
 */
typedef struct {
    Colormap        colormap;
    unsigned        long red_max;
    unsigned        long red_mult;
    unsigned        long green_max;
    unsigned        long green_mult;
    unsigned        long blue_max;
    unsigned        long blue_mult;
}

```

```

        unsigned        long base_pixel;
        VisualID        visualid;        /* added by ICCCM
                                        version 1 */
        XID             killid;         /* added by ICCCM
                                        version 1 */
    } XStandardColormap;
    #define ReleaseByFreeingColormap ((XID) 1L) /*for killid field above */

```

For a more detailed description of standard colormap, see Volume 1, section 7.7, of the *Xlib Programming Manual for Version 11 of the X Window System*, by Adrian Nye, O'Reilly & Associates, Inc.

In summary, the colormap field is the XID of the colormap resource. Each component (red, green, and blue) has a max value and a multiplier. The max value defines the number of shades of that component that will be present in the colour cube. The multiplier value scales each component into the proper range in the 256 entry colormap. The base_pixel defines where, within the colormap resource, the colour cube begins. The visualid defines the visual for the colormap resource. Finally, the killid defines how the colormap resource should be destroyed.

The application should be concerned with the killid field only if it has created the colormap resource itself and, thus, will be concerned with how to destroy it. Section 6.4 of the *Inter-Client Communication Conventions Manual (ICCCM)*, Version 1.0, explains the killid as follows.

When deleting or replacing an RGB_COLOR_MAP, it is not sufficient to delete the property; it is important to free the associated colormap resources as well. If "kill_id" is greater than one, then the resources should be freed by issuing a KillClient protocol request with "kill_id" as the argument. If "kill_id" is one, then the resources should be freed by issuing a FreeColormap protocol request with "colormap" as the Colormap argument. If "kill_id" is zero, then no attempt should be made to free the resources. A client creating an RGB_COLOR_MAP for which the "colormap" resource is created specifically for this purpose should set "kill_id" to one (and can create more than one such standard colormap using a single connection). A client creating an RGB_COLOR_MAP for which the "colormap" resource is shared in some way (e.g., is the default colormap for the root window) should create an arbitrary resource and use its resource id for "kill_id" (and should create no other standard colormaps on the connection).†

† from *Inter-Client Communications Conventions Manual, Version 1.0, MIT X Consortium Standard*, by David S. H. Rosenthal, Sun Microsystems, Inc.

The SunPHIGS default color cube is a 6/9/4 color cube with values as follows:

```

red_max = 5      (6 shades of red)
red_mult = 1
green_max = 8    (9 shades of green)
green_mult = 6

```

```
blue_max = 3    (4 shades of blue)
blue_mult = 54
```

When a workstation is running in remote PEX mode, the default color cube for the PseudoColor visual class is a 4/4/4 color cube with values as follows:

```
red_max = 3     (4 shades of red)
red_mult = 1
green_max = 3   (4 shades of green)
green_mult = 4
blue_max = 3    (4 shades of blue)
blue_mult = 16
```

Conversion of floating point RGB values to indices in the X colormap are performed as follows:

$$I = \text{base_pixel} + \\ ((u_long) (0.5 + (\text{red} * \text{red_max})) * \text{red_mult}) + \\ ((u_long) (0.5 + (\text{green} * \text{green_max})) * \text{green_mult}) + \\ ((u_long) (0.5 + (\text{blue} * \text{blue_max})) * \text{blue_mult})$$

If an application wants to create its own color cube, it should create a colormap resource and a standard colormap property. For a drawable type workstation, the colormap should be associated with the drawable. If the colormap ID in the standard colormap property is not the same as the drawable's colormap ID, the colormap specified in the standard colormap property will still be used if the visual classes correspond. The standard colormap should then be placed on the root window as a property. The PHIGS_X_CMAP_PROP_ATOM workstation type attribute should then be set to the value of the atom representing this property. One limitation is that SunPHIGS performs shading much better if the multipliers are set up such that $\text{red_mult} \leq \text{green_mult} \leq \text{blue_mult}$. If multipliers do not follow this behavior, banding occurs in shaded images.

When the atom specified by the application cannot be used for the workstation, or if no atom is specified, SunPHIGS first looks for RGB_BEST_MAP on the root window. If this cannot be used, it then looks for RGB_DEFAULT_MAP on the root window. If this, too, cannot be used, then it creates its own standard colormap, places it on the root window, and updates PHIGS_X_CMAP_PROP_ATOM appropriately.

If an application's user interface is implemented with the XView toolkit, an XView color map segment should be created in such a way that the PHIGS colour cube can share, if possible, the hardware colormap with the window manager and XView interface objects (window frame, buttons, etc.). One procedure for accomplishing this is to do the following:

1. Set up the desired colours of the colour cube in an Xcolor array
2. Create an XView control color map segment, by creating a dynamic color map segment object (CMS) with CMS_CONTROL_CMS set to TRUE, CMS_SIZE set to the size of the colour cube plus CMS_CONTROL_COLORS, and CMS_X_COLORS set to the address of the Xcolor array containing the

- colour cube. This creates a color map segment containing the control colors used by XView, followed by the colors of the colour cube.
3. Create an XStandardColormap with the colormap created above. The base pixel should be set to the first pixel of the colour cube. Make this standard colormap a property of the root window.
 4. Pass the atom of the above standard colormap property to SunPHIGS by setting the PHIGS_X_CMAP_PROP_ATOM workstation attribute.

This procedure not only minimizes colour flashing between a SunPHIGS workstation and an XView-based user interface, but also makes the colors in the color cube available to XView. The SunPHIGS example program `cmap_panel.c` illustrates the possibilities: it creates a true colour workstation depicting a shaded fill area, and presents an array of coloured XView buttons allowing the user to select colours for the fill area's vertices. Since the colours on the buttons are drawn from the same colormap used by SunPHIGS, there is no flashing as the pointer is moved from the buttons to the workstation window.

Indexed Colour Workstations

On an indexed colour workstation, the application can tell SunPHIGS to share color cells within a given colormap resource among workstations. The workstation type attribute `PHIGS_X_CMAP_PROP_ATOM` gives the application access to a `Phigs_api_index_map` structure stored as a property of the root window. `Phigs_api_index_map` is defined in `phigs.h` as follows:

```
typedef struct {
    Colormap      colormap;    /* XID */
    unsigned int  ncolors;     /* number of cells to share */
    unsigned long base_pixel;  /* offset into colormap */
    VisualID      visualid;    /* visual ID */
    XID           killid;      /* specifies how to destroy
                               colormap */
    unsigned int  count;       /* Used to decide when to delete
                               the property */
} Phigs_api_index_map;
```

If the `PHIGS_X_CMAP_PROP_ATOM` is non-NULL, SunPHIGS looks on the root window for the atom specified by this workstation type attribute and retrieves the `Phigs_api_index_map`. On a drawable type workstation, the drawable's `visualid` must be the same as that specified in the `Phigs_api_index_map`. When the atom specified is NULL, or if it cannot be used, SunPHIGS creates its own colormap and property and places the data on the root window.

In summary, the `colormap` field is the XID of the colormap to be shared. `ncolors` is the number of cells within the X colormap to be shared. `base_pixel` is the offset into the colormap of the first cell for the workstation to use. `visualid` is the Visual ID of the colormap. `killid` specifies how to destroy the colormap and is explained in the *True Colour*

Workstations section, above. *count* determines when to delete the property from the root window. When a workstation decides to use a `Phigs_api_index_map` property, it increments the count by one. When a workstation is closed, the count is decremented. When the count is 0, and the property was created by SunPHIGS at some point, then the property is deleted from the root window.

If the application chooses to create its own colormap resource and wishes to share it among workstations, it must do the following:

1. Create colormap resource.
2. Allocate colors appropriately.
3. Store colors into the colormap.
4. Create a `Phigs_api_index_map` property of type `PHIGS_API_INDEX_MAP`.
5. Initialize property data appropriately, setting count to 0.
6. Place `Phigs_api_index_map` property on root window.
7. Set `PHIGS_X_CMAP_PROP_ATOM` appropriately.

Example:

```
Atom                property_type, property_atom;
Display            *dpy;
Phigs_api_index_map api_index_map;
Pint               ws_type;

/* Either use default colormap or create your own */
/* Allocate your colors */
property_type = XInternAtom(display, PHIGS_API_INDEX_MAP, False);
property_atom = XInternAtom(display, "MY_COLORMAP_ATOM", False);

XChangeProperty(dpy, DefaultRootWindow(dpy), property_atom,
                property_type, 32, PropModeReplace,
                (u_char *)&api_index_map,
                sizeof(Phigs_api_index_map) / 4);

ws_type = phigs_ws_type_create(phigs_ws_type_x_tool,
                              PHIGS_X_CMAP_PROP_ATOM, property_atom,
                              NULL);
```

When a colormap resource is shared among windows, setting a colour representation affects all windows sharing the same colormap resource. If a SunPHIGS workstation colormap is changed by another window's action, then the change is not reflected in the inquiry of its Realized colour representation.

Normally, when the display connection to a particular server is released, all of the colormap properties created by SunPHIGS on that server are destroyed. When `CLOSE PHIGS` is called, all colormap resources created by SunPHIGS will be relinquished. Other applications attempting to use colormap resources created by SunPHIGS should use them

with caution.

If the SunPHIGS program ends abnormally (crashes) or if the programmer fails to close the open workstations on an indexed workstation or to close PHIGS on a true color workstation, then these colormap properties will remain attached to the root window. The names of the SunPHIGS properties currently attached to the root window can be viewed by executing the following command:

```
% sprop -root -notype -len 0 | grep SUNPHIGS | tr -d :=
```

Its output is a list of the SunPHIGS property names in the form of:

```
_SUN_SUNPHIGS_API_CMAP_<pid>_<workstation id>
```

and:

```
_SUN_SUNPHIGS_API_CMAP_<pid>
```

The colormap properties that are left behind have very little negative impact on system performance, aside from each consuming a small amount of memory. They can be removed in the interest of maintaining a clean environment by executing the following command:

```
$ foreach i (`xprop -root -notype -len 0 | grep SUNPHIGS | tr -d :=`
? xprop -root -remove $i
? end
```

**Other Colour
Attributes**

SunPHIGS offers a Generalized Structure Element (GSE) for setting the highlight colour index entry in the PHIGS traversal state list. This entry is used when highlighting subsequent output primitives. The default is the last colour in the colour table. A direct colour cannot be specified for highlighting.

SEE ALSO

INQUIRE COLOUR FACILITIES (3P)
INQUIRE COLOUR MODEL FACILITIES (3P)
INQUIRE DIRECT COLOUR MODEL FACILITIES (3PP)
SET COLOUR MODEL (3P)
SET COLOUR REPRESENTATION (3P)
INQUIRE COLOUR REPRESENTATION (3P)
INQUIRE PREDEFINED COLOUR REPRESENTATION (3P)
SET POLYLINE COLOUR INDEX (3P)
SET POLYLINE COLOUR (3PP)

NAME	INTRO DRAWABLE REGION – the introduction to the <i>X Drawable</i> region workstation type
DESCRIPTION	<p>The <i>X Drawable Region</i> workstation type allows applications increased flexibility in interacting with SunPHIGS. This workstation type allows the definition of workstations as regions in a drawable. These workstations can overlap and have the attribute of <i>not</i> clearing before traversal. Using these qualities, applications can achieve effects not possible using the normal <i>X Drawable</i> workstation type.</p> <p>This workstation type provides powerful and flexible functionality. Each of the drawable region workstations has all of the standard output functionality available to a workstation: control over posted structures, display update state, and so on. SunPHIGS makes no connection among the drawable region workstations, thus leaving the application with the responsibility and flexibility of handling the interaction as it likes. On a drawable that SunPHIGS handles, the only connection among region workstations is the order in which sets of workstations are updated.</p> <p>The term <i>drawable region workstation</i> refers to one area on the screen. The term <i>region workstations on a drawable</i> refers to a set of drawable region workstations or areas on the screen.</p>
Limitations	<p>This workstation type interprets some attributes differently than does the <i>X Drawable</i> workstation type. In particular, some attributes allocate window system or hardware resources on a per-window basis, so that they cannot be mixed among drawable region workstations. For these attributes, the first drawable region workstation sets the values that will be used by the other region workstations opened on the drawable. For example, if the first drawable region workstation specifies hardware double buffering, then all the drawable region workstations (opened on that drawable) will use hardware double buffering. The workstation type attributes that are affected in this way are:</p> <ul style="list-style-type: none"> PHIGS_ACCELERATOR_USAGE PHIGS_X_BUF_MODE PHIGS_COLOUR_MODE PHIGS_COLOUR_TABLE_SIZE PHIGS_X_CMAP_PROP_ATOM <p>All the workstations share a common workstation colour table, so that the setting a colour representation on one drawable region workstation applies to all of the drawable region workstations on the drawable.</p> <p>Drawable region workstations are output only. The workstation category attribute is ignored. Colour map double buffering is not supported on drawable region workstations.</p>
Definition of Regions	<p>Each drawable region workstation draws into an area in its drawable defined by a lower left and upper right corner. These corners are defined in drawable coordinates (pixels), with origin at the lower left hand corner of the drawable. Specifying the default values of</p>

{0,0} for both corners causes the region to occupy the entire drawable.

Workstation Priority

Each drawable region workstation has a priority that is used to determine the order in which the workstations on the drawable are updated when several region workstations are updated at the same time. This can be necessary due to either window damage or use of ESCAPE -18, which is described in the *Update a Set of Region Workstations* section, below. The workstations are updated from lowest to highest priority. If two or more workstations have the same priority, then the order in which they are updated is undefined.

No Background Clear Workstation Attribute

A drawable region workstation can be opened with the *no background clear* attribute, which prevents the background of the workstation from being cleared before each traversal. This allows the workstation to have a *Transparent* background. It is the responsibility of the application to ensure that the background is maintained correctly. The application must use another workstation under the transparent workstation to clear the transparent workstation's drawing surface.

Set Visual Representation State

ESCAPE -17 can be used to set the visual representation state of a drawable region workstation to *Deferred*. This causes the workstation to be redrawn when *update workstation* is called, even if no changes have been made to that workstation. The application can use this escape to handle the interaction between region workstations. For example, if Workstation A is a transparent workstation on top of Workstation B, and the application updates Workstation B (or does an action that causes an implicit regeneration of Workstation B), then the application can use this escape to update Workstation A.

Update a Set of Region Workstations

ESCAPE -18 can be used to update a set of drawable region workstations on a drawable. The effect is the same as if *update workstation* were called for each workstation in the list of workstations. When the workstations are double buffered, using this escape to update a set of workstations may be faster and provide a smoother appearance than using *update workstation*. The workstations are updated in priority order.

Set Region Workstation Rect (ESCAPE -9)

ESCAPE -9 (3P) can be used to change the location and size of a drawable region workstation on a drawable. This causes the workstation to be cleared in the original location (if the workstation is not transparent) and internally updated to reflect the new location and size. There is no traversal at the new location. The visual representation state of the workstation is set to *Deferred*.

SEE ALSO

- ESCAPE -9 (3P)
- ESCAPE -17 (3P)
- ESCAPE -18 (3P)
- WORKSTATION TYPE SET (3P)

NAME	INTRO INTERNATIONALIZATION – introduction to the <i>internationalization</i> extensions to the SunPHIGS graphics library
DESCRIPTION	<p>This document describes features of SunPHIGS for the support of <i>internationalized</i> applications. These applications can be adapted easily to a specific language, or <i>localized</i>. Internationalized applications typically have a core that is constant across locales, and locale-specific modules that contain language-specific information.</p> <p>Areas of SunPHIGS applications that require internationalization are: the character sets used for string device input and echoing, the formatting of numbers and dates in output, the character sets for text output, and the character strings used in error messages. The <i>Solaris</i> operating system provides some support for non-English string input and echoing using the ISO-Latin-1 character set, and for localized formatting of numbers and dates. These localizations are based on environment variables and files that indicate the installation's locale. To enable applications to adapt to locales requiring character sets other than ISO-Latin-1, the internationalization extensions to SunPHIGS provide text output primitives capable of reproducing other character sets.</p>
OVERVIEW	<p>The internationalization extensions provide the ability to switch between different languages or character sets, and to represent and display characters from very large character sets. The extensions are in two main areas: four new GDP primitives, mono-encoded text and mono-encoded annotation text relative, their 3D counterparts, and an ESCAPE that allows the application to inquire the extents of mono-encoded text.</p> <p>This document describes these extensions, and how to convert between mono-encoded text and other text encodings. It also discusses other issues of internationalization, such as error messages, window titles, and other input and output primitives.</p>
DEFINITIONS	The following terms are used throughout this document:
Character Set	Character sets may be composed of alphabets, ideograms or other symbols. ASCII and Kanji (Japanese) are examples of character sets. As a text attribute, the character set determines the graphical characters available for text output, but not their stylistic appearance.
Font	The shapes of the symbols representing each character in a character set. In SunPHIGS, graphical characters in different character sets with the same font will have similar styles. Monospaced, Simplex and Italic are examples of font styles. As a text attribute, the font determines the style in which graphical characters are displayed.
I-String	A list of mono-encodings. It is used to specify strings made up of mixed character sets. It is also used to specify strings where the single character set used is not ASCII.
Mono Encoding	A data record consisting of a NULL-terminated text string, the character set used for that particular text string, and an encoding state.

Encoding State

This holds information that the application can use in converting from a mono-encoding back to an encoding scheme which is stateful. For example, if the application takes EUC encoded text and converts it to an I-String, then the encoding state can be used to store the EUC codeset from which a particular character set was derived. This can be used to convert the I-String back to EUC at a later time. See the section *I-String Conversion* for more details. The encoding state is merely stored in the CSS. It is not actually used by SunPHIGS.

EUC Codeset

A data record consisting of a font and a character set index.

**USING
INTERNATIONALIZED
TEXT**

CHARACTER SETS

The character sets supported by a workstation type can be queried using ESCAPE -10, *inquire character set facilities*. The character sets available are defined as constants in phigs.h and PARAMETER statements in phigs77.h:

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Character Set</i>
0	PCS_ASCII	PCSASCII	ISO-646 (ASCII)
-1	PCS_GREEK	PCSGREEK	Greek
-2	PCS_SYMBOL	PCSSYMBOL	Symbol
-3	PCS_CARTOGRAPHIC	PCSCARTOGRAPHIC	Cartographic
-4	PCS_KANJI	PCSKANJI	JIS-X0208 (Japanese Kanji, formerly JIS-C6226)
-5	PCS_KATAKANA	PCSKATAKANA	Katakana (Japanese Phonetic)

FONTS

Each character set may support different fonts. A list of the fonts and precisions supported by a workstation type for a particular character set can be inquired using ESCAPE -11, *inquire fonts for character set*.

All the font indices have named constants defined in phigs.h and phigs77.h (FORTRAN) as shown below. Note that the constants denote the style of the text, and thus have the same names and values for different character sets.

The ASCII fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced
2	PFONT_SIMPLEX	PFONTSIMPLEX	Simplex
-1	PFONT_DUPLEX	PFONTDUPLEX	Duplex
-2	PFONT_COMPLEX	PFONTCOMPLEX	Complex
-3	PFONT_TRIPLEX	PFONTTRIPLEX	Triplex
-4	PFONT_ITALIC_COMPLEX	PFONTITALICCMPLX	Italic Complex
-5	PFONT_ITALIC_TRIPLEX	PFONTITALICTRPLX	Italic Triplex
-6	PFONT_SCRIPT_SIMPLEX	PFONTSCRIPTSMPLX	Script Simplex
-7	PFONT_SCRIPT_COMPLEX	PFONTSCRIPTCMPLX	Script Complex

The Greek fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced
2	PFONT_SIMPLEX	PFONTSIMPLEX	Simplex
-2	PFONT_COMPLEX	PFONTCOMPLEX	Complex

The symbol fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced
2	PFONT_SIMPLEX	PFONTSIMPLEX	Simplex

The cartographic fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced
2	PFONT_SIMPLEX	PFONTSIMPLEX	Simplex

The Kanji fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced

The Katakana fonts available are as follows:

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Style</i>
1	PFONT_MONO	PFONTMONO	Monospaced

During traversal, if a font is not available in the specified character set, font 1 and that character set will be used. If a specified character set is not available, font 1 and character set 1 will be used.

RELATION TO EXISTING FUNCTIONS

The behavior of existing SunPHIGS functions has been adapted to give compatible behavior when using the internationalization extensions. Specifically, SET TEXT FONT sets the font to be used for all text, regardless of character set. Likewise, the font in SET TEXT REPRESENTATION sets the font entry in the text representation that all text will use, regardless of character set.

The inquiry INQUIRE TEXT EXTENT is used to check the extents of ASCII character strings. ESCAPE -12, *inquire extended text extent*, is provided to perform the same function as INQUIRE TEXT EXTENT using I-Strings instead of simple ASCII character strings.

SunPHIGS supports four GDPs for displaying mono-encoded text:

Mono Encoded Text
 Mono Encoded Annotation Text Relative
 Mono Encoded Text 3
 Mono Encoded Annotation Text Relative 3

These four GDPs provide the extended support for the TEXT, ANNOTATION TEXT RELATIVE, TEXT 3, and ANNOTATION TEXT RELATIVE 3 primitives respectively.

I-STRING CONVERSION

This section describes how mono-encoded text can be converted to other encoding schemes such as EUC and vice verse.

EUC encoded text has four *codesets*, where each codeset consists of a character set and a font index. Every character in a text string is encoded in such a way as to describe the codeset to which it belongs. For converting to and from EUC we can define the following constants:

ENCODING_MASK	= 0xF0	<i>For looking at only the encoding information</i>
ENCODING_EUC	= 0x00	<i>Encoding ID for Extended UNIX Coding</i>
CODESET_MASK	= 0x0F	<i>For looking at only the state information</i>
CODESET_EUC_CS0	= 0x00	<i>The character set is in code Set 0</i>
CODESET_EUC_CS1	= 0x01	<i>The character set is in code Set 1</i>
CODESET_EUC_CS2	= 0x02	<i>The character set is in code Set 2</i>
CODESET_EUC_CS3	= 0x03	<i>The character set is in code Set 3</i>

Now, to convert EUC encoded text to an I-string we first need to break the text into contiguous strings of characters belonging to one codeset. Each of these substrings will be a mono-encoding. With each mono-encoding we will store the substring, and, looking at the codeset for that substring, we can get and store the character set for the substring. Each mono-encoding also has its own encoding state, which we can use to store both the fact that this mono encoding was derived from EUC encoded text, and we can store the codeset from which the character set was derived. The list of mono-encodings will form an I-string. We can then use the font in codeset 0 as input to SET TEXT FONT. The only limitation to this is in the case where different codesets have different fonts. An I-string may only use one font.

Converting from the I-string back to EUC encoded text is a simple matter of looking at the encoding state for each mono-encoding to determine the codeset from which that substring was derived.

The examples directory has three associated files: *mono_text_utils.c*, *mono_text_utils.h*, and *euc_text.c*. *mono_text_utils.c* contains a set of utilities for converted encoded text to, and from, mono-encoding. Currently, only EUC conversion is supported in this file. *mono_text_utils.h* is a support file that should be included by any application using the utilities in *mono_text_utils.c*. *euc_text.c* is an example application that uses the

**OTHER
INTERNATIONALIZATION
ISSUES**

conversion utilities in *mono_text_utils.c*.

SunPHIGS Error Messages:

The *Sun Internationalization and Localization Guide* describes the standard library functions for internationalizing application programs. To assist the application developer, SunPHIGS uses these library functions for the output of PHIGS error messages, allowing the text and syntax of these messages to be localized.

The message libraries for SunPHIGS are found, by default, under:

`$(PHIGSHOME)/lib/locale/locale-name/LC_MESSAGES`

and are contained in the following three files:

`SUNW_PHIGS_ERR.mo`

containing the text of specific standardized error messages.

`SUNW_PHIGS_FUN.mo`

containing the names of functions.

`SUNW_PHIGS_MSG.mo`

containing the format strings used in error reporting, and a small set of miscellaneous message strings.

The source files for these pre-compiled message libraries can be found in the same directory as the SunPHIGS include files, with a `.po` extension; these source files can be used as the basis for a translation of SunPHIGS error messages into another locale, if a translation for that locale is not otherwise available. The default locale for SunPHIGS error messages is the "en" locale, rather than the "C" locale, due to an unfortunate optimization in the Solaris 2.x version of the internationalization library functions, in which the message library files are ignored in the "C" domain. If the current domain is "C", SunPHIGS will change it internally to "en"; this should not effect application programs expecting the "C" domain, because their message library files would be ignored anyway.

Window and Icon Labels:

For the default icon and window labels of X Tool workstations, SunPHIGS uses the standard X Window system resource database mechanism. These labels can also be set using the workstation type attributes `PHIGS_TOOL_LABEL` and `PHIGS_TOOL_ICON_LABEL` (see the `WORKSTATION TYPE SET (3PP)` man page).

Other Input and Output Functions:

The SunPHIGS internationalization extensions provide text and annotation text primitives capable of supporting a variety of character sets. The other text output and input functions of SunPHIGS, however, are limited by the underlying software to accepting only the ISO-Latin-1 character set. These include the `MESSAGE` output function, the prompts and labels associated with input device types such as `CHOICE` and `VALUATOR`, and the input that can be accepted by the `STRING` input device. With the current version of SunPHIGS, an application wishing to use other character sets for input and output (other than text or annotation text) should implement its input and output using another programming toolkit, along with an X Drawable or X Drawable Region workstation.

SEE ALSO

ESCAPE -10 (3P)
ESCAPE -11 (3P)
ESCAPE -12 (3P)
GENERALIZED DRAWING PRIMITIVE -17 (3P)
GENERALIZED DRAWING PRIMITIVE -18 (3P)
GENERALIZED DRAWING PRIMITIVE 3 -17 (3P)
GENERALIZED DRAWING PRIMITIVE 3 -18 (3P)
Software Internationalization Guide

NAME PHIGS DESCRIPTION TABLE – contains information describing the static capabilities of SunPHIGS

DESCRIPTION The PHIGS Description Table is created when PHIGS is opened and exists until PHIGS is closed. The PHIGS standard includes the default values for the traversal state list in the logical PHIGS Description Table. These values are listed in the PHIGS TRAVERSAL STATE LIST (7P).

The remaining values are limits and capabilities. These may be queried for.

Data Type Abbreviations
 I Integer W
 E Enumeration Type L(type)

SunPHIGS Description Table

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
number of available workstation types	I	4
list of available workstation types	L(W)	see Table 1, PHIGS WORKSTATION DESCRIPTION TABLE (7P)
max number of simultaneously open workstations	I	20
max number of simultaneously open archive files	I	25
number of available names for name sets	I	1024
number of available character sets	I	†
— list of available character sets	I	†
max length of normal filter list for ISS	I	MAXINT (unlimited)
max length of inverted filter list for ISS	I	MAXINT (unlimited)
number of available generalized structure elements	I	9
— list of available generalized structure elements		see Tables 1 and 2
max number of distinct planes in modelling clip volumes	I	8
number of available modelling clip operators	I	2
— modelling clipping operator	I	1 (replace)
— modelling clipping operator	I	2 (intersect)

† See INTRO INTERNATIONALIZATION (7P).

Table 1 — Available Generalized Structure Elements

<i>GSE Id</i>	<i>Name</i>	<i>Workstation Dependency Indicator</i>
-1	Highlight Color	WORKSTATION INDEPENDENT
-4	Text Slant Angle	WORKSTATION INDEPENDENT
-5	Annotation Text Slant Angle	WORKSTATION INDEPENDENT
-6	Polygon Type	WORKSTATION INDEPENDENT
-7	Anti-aliasing Identifier	WORKSTATION INDEPENDENT
-8	Text Stroke Width Scale Factor	WORKSTATION INDEPENDENT
-9	Stroke End Cap	WORKSTATION INDEPENDENT
-10	Stroke Join	WORKSTATION INDEPENDENT
-11	Silhouette Edge Flag	WORKSTATION INDEPENDENT

Table 2 — Available Generalized Structure Elements

<i>GSE Id</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Meaning</i>
-1	PUGSE_HIGHLIGHT_COLOUR_INDEX	PUGSEHLCI	Set Highlight Colour Index
-4	PUGSE_TEXT_SLANT_ANGLE	PUGSESETTSA	Set Text Slant Angle
-5	PUGSE_ATEXT_SLANT_ANGLE	PUGSESETATSA	Set Annotation Text Slant Angle
-6	PUGSE_PGON_TYPE	PUGSEPGONTYPE	Set Polygon Type
-7	PUGSE_SET_STROKE_AA_ID	PUGSESETAAID	Set Stroke Anti-aliasing Identifier
-8	PUGSE_SET_STROKE_WIDTH_SCALE_FACTOR	PUGSESETTXSWSF	Set Text Stroke Width Scale Factor
-9	PUGSE_SET_STROKE_END_CAP	PUGSESETSTEC	Set Stroke End Cap
-10	PUGSE_SET_STROKE_JOIN	PUGSESETSTJ	Set Stroke Join
-11	PUGSE_SIL_EDGE_FLAG	PUGSESELEDGEF	Set Silhouette Edge Flag

SEE ALSO

PHIGS TRAVERSAL STATE LIST (7P)

PHIGS WORKSTATION DESCRIPTION TABLE (7P)

NAME	PHIGS TRAVERSAL STATE LIST – contains all transient information SunPHIGS uses during traversal			
DESCRIPTION	<p>The PHIGS Traversal State List is created whenever a traversal is initiated, and only exists during the traversal. Hence, its values cannot be inquired for. Traversal is the execution of a structure network. Traversal begins with a structure posted to a workstation. Structure elements are traversed sequentially. The traversal state list is modified by executing attribute elements.</p> <p>The initial values in the traversal state list are defined by the PHIGS standard or implementation-dependent. These values are inherited by a posted structure. Whenever an EXECUTE STRUCTURE element is traversed, the executed structure, or <i>child</i> structure, <i>inherits</i> the attributes of the <i>parent</i> structure; that is, the structure invoking (executing or referencing) it. Different parent structures can assign different attribute values.</p> <p>The <i>child</i> structure's attributes are then modified by attribute elements as they are sequentially executed. The child structure may also execute subordinate structures, at which point the calling structure is considered the parent of this generation. At the end of the child structure (that is, after its last element), traversal returns to the parent structure, restoring the saved state of the parent structure, and continues until the end of the parent structure. This continues until the entire structure network is traversed (that is, until traversal of the posted structure is complete). The modifications to the child's structure do not affect the parent structure's attributes. Traversal state list values may be said to <i>expire</i> at the end of the structure in which they are set, as traversal of the parent structure resumes.</p> <p>The values in the traversal state list are <i>bound</i> to primitives as the primitive elements are sequentially executed.</p>			
Data Type Abbreviations	I	Integer	GCOLR	General Colour†
	R	Real	V2/V3	2D/3D Vector
	E	Enumeration Type	P3	3D Point
	L(type)	List of values of a given type	SET(NM)	Set of Eligible Names
	MCV	Modelling Clipping Volume	(nys)	Not Yet Supported

**SunPHIGS Traversal
State List**

<i>Traversal State List Entry</i>	<i>Data Type</i>	<i>Default or Initial Value</i>
polyline index	I	1
linetype	I	1
linewidth scale factor	R	1.0
polyline colour index	I	1
polyline colour†	GCOLR	(INDIRECT, 1)
linetype ASF	E	INDIVIDUAL
linewidth scale factor ASF	E	INDIVIDUAL
polyline colour index ASF	E	INDIVIDUAL
polyline shading method†	I	1 (NONE)
curve approximation criteria type†	I	2 (Constant Parametric Between Knots)
curve approximation criteria value†	R	1.0
polyline shading method ASF†	E	INDIVIDUAL
curve approximation criteria ASF†	E	INDIVIDUAL
polyline anti-aliasing identifier (GSE -7)‡	E	none
polyline wide stroke end-cap (GSE -9)‡	E	BUTT
polyline wide stroke join (GSE -10)‡	E, R	BUTT (threshold unused)
polymarker index	I	1
marker type	I	3
marker size scale factor	R	1.0
polymarker colour index	I	1
polymarker colour†	GCOLR	(INDIRECT, 1)
marker type ASF	E	INDIVIDUAL
marker scale factor ASF	E	INDIVIDUAL
polymarker colour index ASF	E	INDIVIDUAL
polymarker anti-aliasing identifier (GSE -7)‡	E	none
text index	I	1
text font	I	1 (Monospaced Roman Simplex)
text precision	E	STRING
character expansion factor	R	1.0
character spacing	R	0.0
text colour index	I	1
text colour†	GCOLR	(INDIRECT, 1)
text font ASF	E	INDIVIDUAL
text precision ASF	E	INDIVIDUAL
character expansion factor ASF	E	INDIVIDUAL

This table continues on the next page

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

‡ This is a SunPHIGS-specific item of traversal state. See the appropriate GSE man page.

SunPHIGS Traversal
State List (continued)

<i>Traversal State List Entry</i>	<i>Data Type</i>	<i>Default or Initial Value</i>
character spacing ASF	E	INDIVIDUAL
text colour index ASF	E	INDIVIDUAL
character height	R	0.01
character up vector	V2	(0.0,1.0)
character width	R	0.01
character base vector	V2	(1.0,0.0)
text path	E	RIGHT
text alignment (horz & vert)	2×E	(NORMAL, NORMAL)
text slant angle (GSE -4) †	R	0.0
text anti-aliasing identifier (GSE -7) †	E	none
text stroke width scale factor (GSE -8) †	R	1.0
text wide stroke end-cap (GSE -9) †	E	BUTT
text wide stroke line-join (GSE -10) †	E, R	BUTT (threshold unused)
annotation text character height	R	0.01
annotation text character up vector	V2	(0.0,1.0)
annotation text character width	R	0.01
annotation text character base vector	V2	(1.0,0.0)
annotation text path	E	RIGHT
annotation text alignment (horz & vert)	2×E	(NORMAL, NORMAL)
annotation text slant angle (GSE -5) †	R	0.0
annotation style	I	1 (unconnected)
interior index	I	1
interior style	E	HOLLOW
back interior style †	E	HOLLOW
interior style index	I	1
back interior style index †	I	1
interior colour index	I	1
interior colour †	GCOLR	(INDIRECT, 1)
back interior colour †	GCOLR	(INDIRECT, 1)
interior style ASF	E	INDIVIDUAL
back interior style ASF †	E	INDIVIDUAL
interior style index ASF	E	INDIVIDUAL
back interior style index ASF †	E	INDIVIDUAL
interior colour index ASF	E	INDIVIDUAL

This table continues on the next page

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

‡ This is a SunPHIGS-specific item of traversal state. See the appropriate GSE man page.

SunPHIGS Traversal
State List (continued)

<i>Traversal State List Entry</i>	<i>Data Type</i>	<i>Default or Initial Value</i>
back interior colour ASF†	E	INDIVIDUAL
interior anti-aliasing identifier (GSE -7)‡	E	none
ambient reflection coefficient†	R	1.0
back ambient reflection coefficient†	R	1.0
diffuse reflection coefficient†	R	1.0
back diffuse reflection coefficient†	R	1.0
specular reflection coefficient†	R	1.0
back specular reflection coefficient†	R	1.0
specular colour†	GCOLR	(RGB, WHITE)
back specular colour†	GCOLR	(RGB, WHITE)
specular exponent†	R	0.0
back specular exponent†	R	0.0
transparency coefficient†*	R	0.0 (opaque)
back transparency coefficient†*	R	0.0 (opaque)
interior shading method†	I	1 (NONE)
back interior shading method†	I	1 (NONE)
interior reflectance equation†	I	1 (NONE)
back interior reflectance equation†	I	1 (NONE)
area properties ASF†	E	INDIVIDUAL
back area properties ASF†	E	INDIVIDUAL
interior shading method ASF†	E	INDIVIDUAL
back interior shading method ASF†	E	INDIVIDUAL
interior reflectance equation ASF†	E	INDIVIDUAL
back interior reflectance equation ASF†	E	INDIVIDUAL
surface approximation criteria type†	I	2 (Constant Parametric Between Knots)
surface approximation criteria value†	2×R	(1.0,1.0)
surface approximation criteria ASF†	E	INDIVIDUAL
parametric surface characteristics type	I	1 (NONE)
parametric surface characteristics record	Record	(empty)
face distinguishing mode†	E	(NO)
face culling mode†	E	(NONE)
depth cue index†	I	0

This table continues on the next page

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

‡ This is a SunPHIGS-specific item of traversal state. See the appropriate GSE man page.

* Only GT-accelerated workstations implement transparency. See ESCAPE -43 (3P).

SunPHIGS Traversal
State List (continued)

<i>Traversal State List Entry</i>	<i>Data Type</i>	<i>Default or Initial Value</i>
colour mapping index†	I	0
rendering colour model†	I	0 (nys)*
light source state†	L(I)	(empty)
edge index	I	1
edge flag	E	OFF
edgetype	I	1
edgewidth scale factor	R	1.0
edge colour index	I	1
edge colour†	GCOLR	(INDIRECT, 1)
edge flag ASF	E	INDIVIDUAL
edgetype ASF	E	INDIVIDUAL
edgewidth scale factor ASF	E	INDIVIDUAL
edge colour index ASF	E	INDIVIDUAL
edge anti-aliasing identifier (GSE -7)‡	E	(NONE)
edge wide stroke end-cap (GSE -9)‡	E	BUTT
edge wide stroke line-join (GSE -10)‡	E, R	BUTT (threshold unused)
silhouette edge flag (GSE -11)‡	E	OFF
pattern size	2×R	(1.0,1.0) (nys)
pattern reference point	P3	(0.0,0.0,0.0) (nys)
pattern reference vectors	2×V3	(1.0,0.0,0.0) (0.0,1.0,0.0) (nys)
pick identifier	I	0
view index	I	0
HLHSR identifier	I	0
name set	SET(NM)	no classes (empty set)
highlight colour index (GSE -1)‡	I	(MAX COL INDEX VALUE)
global modelling transformation	4×4×R	identity
local modelling transformation	4×4×R	identity
modelling clipping volume	MCV	all of WC
modelling clipping indicator	E	NOCLIP

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

‡ This is a SunPHIGS-specific item of traversal state. See the appropriate GSE man page.

* Zero is workstation dependent.

SEE ALSO

PHIGS DESCRIPTION TABLE (7P)

EXECUTE STRUCTURE (3P)

INQUIRE PATHS TO ANCESTORS (3P)

INQUIRE PATHS TO DESCENDANTS (3P)
ELEMENT SEARCH (3P)

NAME PHIGS WORKSTATION DESCRIPTION TABLE – contains information describing the static capabilities of a SunPHIGS workstation

Data Type Abbreviations

I	Integer	P3	3D Point
R	Real	FP	Font/Precision Pair
E	Enumeration Type	CC	Chromaticity Coefficient
B	Bounding Range	D	Data Record
L(type)	List of values of a given type	(nys)	Not Yet Supported

SunPHIGS Workstation Description Table

<i>Workstation Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
workstation type		See Table 1
workstation category		See Table 1
device coordinate units	E	OTHER
maximum display space size:		
in device coordinates	3×R	See Table 26
in device address units	3×I	See Table 26
number of available HLHSR identifiers	I	2
list of available HLHSR identifiers	L(I)	OFF, ON
number of available HLHSR modes	I	3
list of available HLHSR modes	L(I)	NONE, ZBUFF, ZBUFF_NO_ID
number of predefined view representations	I	6
table of predefined view representations		See Table 2
raster or vector display	E	RASTER (OTHER for CGM Output)
dynamic modification accepted for:		
view representation	E	IRG (Implicit Regeneration Required)
polyline bundle representation	E	IRG
polymarker bundle representation	E	IRG
text bundle representation	E	IRG
interior bundle representation	E	IRG
edge bundle representation	E	IRG
pattern representation	E	IRG
colour representation	E	IMM‡ (Immediate)

This table continues on the next page.

‡ IRG for CGM Output workstation type or when in true colour mode — either on a true colour device, or because of the PHIGS Colour Mode attribute — See WORKSTATION TYPE SET(3P).

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
workstation transformation	E	IRG
highlighting filter	E	IRG
invisibility filter	E	IRG
HLHSR mode	E	IRG
default value for display update state:		
deferral mode	E	ASAP (WAIT for CGM Output)
modification mode	E	UQUM (NIVE for CGM Output)
number of available linetypes	I	8
list of available linetypes	L(I)	See Table 3
number of available linewidths	I	0 (continuous)
nominal linewidth	R	1.0 (pixel)
minimum linewidth	R	1.0
maximum linewidth	R	1000.0
number of predefined polyline bundles	I	8
table of predefined polyline bundles		See Table 4
number of available polyline end caps	I	1 (3 on GT)
list of available polyline end caps		See Table 27
number of available polyline join styles	I	1 (4 on GT)
list of available polyline join styles		See Table 28
number of polyline shading methods supported†	I	2
list of polyline shading methods supported†		See Table 18
number of curve approximation criteria types supported†	I	6
list of curve approximation criteria supported†		See Table 22
maximum non-uniform b-spline curve order supported†	I	10
maximum uniform parametric polynomial curve order supported†	I	0 (nys)
number of parametric polynomial curve types supported†	I	0 (nys)
list of parametric polynomial curve types†		(nys)
number of available marker types	I	8
list of available marker types	L(I)	See Table 5

This table continues on the next page.

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
minimum marker size	R	1.0
number of available marker sizes	I	0 (continuous)
nominal marker size	R	11.0 (pixels)
maximum marker size	R	unlimited
number of predefined polymarker bundles	I	6
table of predefined polymarker bundles		See Table 6
number of text font and precision pairs	I	27
list of text font and precision pairs	L(FP)	See Table 7
number of available character expansion factors	I	0 (continuous)
minimum character expansion factor	R	0.0
maximum character expansion factor	R	unlimited
number of available character heights	I	0 (continuous)
minimum character height	R	0.0
maximum character height	R	unlimited
number of predefined text bundles	I	9
table of predefined text bundles		See Table 8
number of available text end caps	I	1 (3 on GT)
list of available text end caps		See Table 27
number of available text join styles	I	1 (4 on GT)
list of available text join styles		See Table 28
number of available annotation styles	I	2
list of available annotation styles	L(I)	See Table 14
number of available interior styles	I	4
list of available interior styles	L(E)	See Table 9
number of available hatch styles	I	24
list of available hatch styles	L(I)	See Table 10
number of predefined interior bundles	I	5
table of predefined interior bundles		See Table 11
number of interior shading methods†	I	2
list of interior shading methods†		See Table 19
number of reflectance equations supported†	I	4
list of reflectance equations supported†		See Table 21
number of surface approximation criteria supported†	I	6
list of surface approximation criteria supported†		See Table 23

This table continues on the next page.

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
maximum uniform parametric polynomial surface	I	0 (nys)
maximum non-uniform b-spline surface order supported†	I	10
maximum trimming curve order supported†	I	6
number of parametric polynomial surface types supported†	I	(nys)
list of parametric polynomial surface types†		(nys)
number of predefined pattern representations	I	0 (not supported)
table of predefined pattern representations		(none)
number of available colour models	I	1
list of available colour models	L(I)	1 (RGB)
number of available edgetypes	I	8
list of available edgetypes	L(I)	See Table 3
number of available edgewidths	I	0 (continuous)
nominal edgewidth	R	1.0 (pixel)
minimum edgewidth	R	1.0
maximum edgewidth	R	1000.0
number of predefined edge bundles	I	8
table of predefined edge bundles		See Table 12
number of available edge end caps	I	1 (3 on GT)
list of available edge end caps		See Table 27
number of available edge join styles	I	1 (4 on GT)
list of available edge join styles		See Table 28
default colour model	I	1 (RGB)
chromaticity coefficients of display primaries	3×CC	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
number of available colours or intensities	I	2 ²⁴
colour available	E	COLOUR
number of predefined colour representations	I	8
table of predefined colour representations		See Table 13
number of available generalized drawing primitives 3	I	17
list of available generalized drawing primitives 3		See Table 15
list of attribute sets used by generalized drawing primitives 3	L(E)	See Table 17

This table continues on the next page.

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
list of attribute sets used by number of available generalized drawing primitives	L(E)	See Table 17
number of available generalized drawing primitives	I	17
list of available generalized drawing primitives		See Table 16
generalized drawing primitives		See Table 16
number of available generalized structure elements	I	0♥
list of available generalized structure elements		none♥
number of user-defined polymarkers	I	0
list of user-defined polymarkers	L(L(P3))	(none)
decomposition flag	E	0●
number of display priorities supported	I	0 (unlimited)
maximum number of polyline bundle table entries	I	1024
maximum number of polymarker bundle table entries	I	1024
maximum number of text bundle table entries	I	1024
maximum number of interior bundle table entries	I	1024
maximum number of edge bundle table entries	I	1024
maximum number of pattern table entries	I	0 (not supported)
maximum number of colour indices	I	32●
maximum number of view indices	I	64●
dynamic modification accepted for:		
structure content modifications	E	CBS (IRG for CGM Output)
post structure	E	CBS (IRG for CGM Output)
unpost structure	E	CBS (IRG for CGM Output)
delete structure	E	CBS (IRG for CGM Output)
reference modifications	E	IRG

This table continues on the next page.

♥ No workstation-dependent generalized structure elements are supported; See PHIGS DESCRIPTION TABLE (7P) for workstation-independent generalized structure elements.

● Settable by using WORKSTATION TYPE SET (3P).

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
number of predefined depth cue indices†	I	2
table of predefined depth cue operations†		See Table 24
maximum number of depth cue bundle table entries†	I	1024
number of predefined colour mapping table entries†	I	(nys)
table of predefined colour mappings†		(nys)
maximum number of colour mapping table entries†	I	(nys)
number of colour mapping methods supported†	I	(nys)
list of colour mapping methods supported†		(nys)
number of available true colours†	I	2 ²⁴
number of available pseudo colours†	I	(nys)
number of directly specifiable colour models supported†	I	1 (RGB)
number of rendering colour models supported†	I	(nys — RGB is used)
number of light source types supported†	I	4
list of light source types supported†		See Table 20
number of predefined light source table indices†	I	5
table of predefined light sources†		See Table 25
maximum number of light source table entries†	I	32
maximum number of simultaneously active† non-ambient light sources	I	32
dynamic modification accepted for:		
all extended bundle representations†	E	IRG
light source representation†	E	IRG
depth cue table entry†	E	IRG
colour mapping representation†	E	IRG
number of logical devices of class LOCATOR	I	5
for every logical input device of class LOCATOR:		
locator device number	I	See Initialize Locator
default initial locator position	P3	See Initialize Locator
number of available prompt and echo types	I	See Initialize Locator
list of available prompt and echo types	L(I)	See Initialize Locator

This table continues on the next page.

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
default echo volume	3×B	See Initialize Locator
default locator data record	D	See Initialize Locator
number of logical input devices of class STROKE for every logical input device of class STROKE:	I	3
stroke device number	I	See Initialize Stroke
maximum input buffer size	I	See Initialize Stroke
number of available prompt and echo types	I	See Initialize Stroke
list of available prompt and echo types	L(I)	See Initialize Stroke
default echo volume	3×B	See Initialize Stroke
default stroke data record containing at least:	D	See Initialize Stroke
input buffer size	I	See Initialize Stroke
number of logical input devices of class VALUATOR for every logical input device of class VALUATOR:	I	18*
valuator device number	I	See Initialize Valuator
default initial value	R	See Initialize Valuator
number of available prompt and echo types	I	See Initialize Valuator
list of available prompt and echo types	L(I)	See Initialize Valuator
default echo volume	3×B	See Initialize Valuator
default valuator data record containing at least:	D	See Initialize Valuator
low value	R	See Initialize Valuator
high value	R	See Initialize Valuator
number of logical input devices of class CHOICE for every logical input device of class CHOICE:	I	7**
choice device number	I	See Initialize Choice
maximum number of choice alternatives	I	See Initialize Choice
number of available prompt and echo types	I	See Initialize Choice
list of available prompt and echo types	L(I)	See Initialize Choice
default echo volume	3×B	See Initialize Choice
default choice data record	D	See Initialize Choice
number of logical input devices of class PICK for every logical input device of class PICK:	I	6
pick device number	I	See Initialize Pick
number of available prompt and echo types	I	See Initialize Pick
list of available prompt and echo types	L(I)	See Initialize Pick

This table continues on the next page.

* 8 valuator are hardware dials, available if a Sun dialbox is attached.

** Device 7 is a hardware button box, available if a Sun button box is attached.

**SunPHIGS
Workstation
Description Table
(continued)**

<i>Description Table Entry</i>	<i>Data Type</i>	<i>Initial Value</i>
default echo volume	3×B	See Initialize Pick
default pick data record containing at least:	D	See Initialize Pick
pick path order	E	TOP_FIRST
number of logical input devices of class STRING	I	1
for every logical input device of class STRING:		
string device number	I	See Initialize String
maximum input buffer size	I	See Initialize String
number of available prompt and echo types	I	See Initialize String
list of available prompt and echo types	L(I)	See Initialize String
default echo volume	3×B	See Initialize String
default string data record containing at least:	D	See Initialize String
input buffer size (characters)	I	See Initialize String
initial cursor position	I	See Initialize String

Inquiry Functions

The following functions may be used to inquire workstation description table values at runtime for programmatic adaptation to the workstation type:

- INQUIRE LIST OF AVAILABLE WORKSTATION TYPES (3P)
- INQUIRE WORKSTATION CATEGORY (3P)
- INQUIRE DISPLAY SPACE SIZE (3P)
- INQUIRE HLHSR FACILITIES (3P)
- INQUIRE VIEW FACILITIES (3P)
- INQUIRE PREDEFINED VIEW REPRESENTATION (3P)
- INQUIRE WORKSTATION CLASSIFICATION (3P)
- INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES (3P)
- INQUIRE DEFAULT DISPLAY UPDATE STATE (3P)
- INQUIRE POLYLINE FACILITIES (3P)
- INQUIRE POLYLINE FACILITIES PLUS (3PP)"
- INQUIRE PREDEFINED POLYLINE REPRESENTATION (3P)
- INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS (3PP)"
- INQUIRE CURVE AND SURFACE FACILITIES (3PP)"
- INQUIRE POLYMARKER FACILITIES (3P)
- INQUIRE PREDEFINED POLYMARKER REPRESENTATION (3P)
- INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS (3PP)"
- INQUIRE TEXT FACILITIES (3P)
- INQUIRE PREDEFINED TEXT REPRESENTATION (3P)
- INQUIRE PREDEFINED TEXT REPRESENTATION PLUS (3PP)"
- INQUIRE ANNOTATION FACILITIES (3P)
- INQUIRE INTERIOR FACILITIES (3P)
- INQUIRE INTERIOR FACILITIES PLUS (3PP)"
- INQUIRE PREDEFINED INTERIOR REPRESENTATION (3P)

INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS (3PP)"
 INQUIRE PATTERN FACILITIES (3P)
 INQUIRE PREDEFINED PATTERN REPRESENTATION (3P)
 INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS (3PP)"
 INQUIRE EDGE FACILITIES (3P)
 INQUIRE PREDEFINED EDGE REPRESENTATION (3P)
 INQUIRE PREDEFINED EDGE REPRESENTATION PLUS (3PP)"
 INQUIRE COLOUR MODEL FACILITIES (3P)
 INQUIRE COLOUR FACILITIES (3P)
 INQUIRE PREDEFINED COLOUR REPRESENTATION (3P)
 INQUIRE GENERALIZED DRAWING PRIMITIVE 3 (3P)
 INQUIRE GENERALIZED DRAWING PRIMITIVE (3P)
 INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES (3P)
 INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED (3P)
 INQUIRE WORKSTATION STATE TABLE LENGTHS (3P)
 INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS (3PP)"
 INQUIRE DYNAMICS OF STRUCTURES (3P)
 INQUIRE DEPTH CUE FACILITIES (3PP)"
 INQUIRE PREDEFINED DEPTH CUE REPRESENTATION (3PP)"
 INQUIRE DIRECT COLOUR MODEL FACILITIES (3PP)"
 INQUIRE LIGHT SOURCE FACILITIES (3PP)"
 INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION (3PP)"
 INQUIRE PREDEFINED INTERIOR REPRESENTATION (3PP)"
 INQUIRE PREDEFINED GENERAL INTERIOR REPRESENTATION (3PP)"
 INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES (3P)
 INQUIRE DEFAULT LOCATOR DEVICE DATA 3 (3P)
 INQUIRE DEFAULT STROKE DEVICE DATA 3 (3P)
 INQUIRE DEFAULT VALUATOR DEVICE DATA 3 (3P)
 INQUIRE DEFAULT CHOICE DEVICE DATA 3 (3P)
 INQUIRE DEFAULT PICK DEVICE DATA 3 (3P)
 INQUIRE DEFAULT STRING DEVICE DATA 3 (3P)

**Table 1 —
Workstation Type
and Category**

<i>Type</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Category</i>
X Tool	phigs_ws_type_x_tool	phigswsttool	OUTIN
X Drawable	phigs_ws_type_x_drawable	phigswstdraw	OUTPUT
CGM Output	phigs_ws_type_cgm_out	phigswstcgmount	MO
X Drawable Region	phigs_ws_type_x_drawable_region	phigswstdrawregion	OUTPUT

Table 2 — Predefined View Representations

<i>View</i>	<i>Orientation Matrix</i>	<i>Mapping Matrix</i>	<i>Clip Limits †</i>
0 (full view)	$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.0, 1.0] [0.0, 1.0] [0.0, 1.0]
1 (top view)	$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.25 \\ 0.0 & 0.1 & 0.0 & 0.75 \\ 0.0 & 0.0 & 0.25 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.05, 0.45] [0.55, 0.95] [0.0, 1.00]
2 (off axis view)	$\begin{bmatrix} 0.7071 & 0.0 & -0.7071 & 0.0 \\ -0.4083 & 0.8165 & -0.4083 & 0.0 \\ 0.5774 & 0.5774 & 0.5774 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.75 \\ 0.0 & 0.1 & 0.0 & 0.75 \\ 0.0 & 0.0 & 0.25 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.55, 0.95] [0.55, 0.95] [0.0, 1.0]
3 (front view)	$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.25 \\ 0.0 & 0.1 & 0.0 & 0.25 \\ 0.0 & 0.0 & 0.25 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.05, 0.45] [0.05, 0.45] [0.0, 1.0]
4 (right side view)	$\begin{bmatrix} 0.0 & 0.0 & -1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.75 \\ 0.0 & 0.1 & 0.0 & 0.25 \\ 0.0 & 0.0 & 0.25 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.55, 0.95] [0.05, 0.45] [0.0, 1.0]
5 (bottom view)	$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & -1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.1 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.25 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$	[0.3, 0.7] [0.3, 0.7] [0.0, 1.0]

† The clip indicator for x-y, back, and front clipping planes is CLIP for all views.

**Table 3 — Available
Line and/or Edge
Types**

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
1	PLINE_SOLID	PLSOLI	Solid
2	PLINE_DASH	PLDASH	Dashed
3	PLINE_DOT	PLDOT	Dotted
4	PLINE_DASH_DOT	PLDASD	Dash-dotted
0	PLINE_LONG_DASH	PLNLONGDASH	Long-dashed
-1	PLINE_DOT_DASH_DOT	PLNDOTDASHDDOT	Dot-dashed-dot-dotted
-2	PLINE_CENTER	PLNCENTER	Center (long-short dashed)
-3	PLINE_PHANTOM	PLNPHANTOM	Phantom (long-short-short dashed)

**Table 4 — Predefined
Extended† Polyline
Bundle Table**

<i>Bundle Index</i>	<i>Linetype</i>	<i>Line Width Scale Factor</i>	<i>Colour Index‡</i>	<i>Shading Method†</i>	<i>Approximation Type†</i>	<i>Approx. Value†</i>
1	Solid	1.0	1	None	WS dependent	5.0
2	Dashed	1.0	1	None	WS dependent	5.0
3	Dotted	1.0	1	None	WS dependent	5.0
4	Dash-dotted	1.0	1	None	WS dependent	5.0
5	Long-dashed	1.0	1	None	WS dependent	5.0
6	Dot-dashed-dot-dotted	1.0	1	None	WS dependent	5.0
7	Center	1.0	1	None	WS dependent	5.0
8	Phantom	1.0	1	None	WS dependent	5.0

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard.

‡ All Predefined Extended Polyline Bundle entries use colour model INDIRECT.

**Table 5 — Available
Marker Types**

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
1	PMARKER_DOT	PPOINT	Point
2	PMARKER_PLUS	PPLUS	Plus
3	PMARKER_ASTERISK	PAST	Asterisk
4	PMARKER_CIRCLE	POMARK	Circle
5	PMARKER_CROSS	PXMARK	Cross
0	PMARKER_SQUARE	PSQUARE	Square
-1	PMARKER_BOWTIE_NE	PBOWTIENE	Bow tie NE/SW
-2	PMARKER_BOWTIE_NW	PBOWTIENW	Bow tie NW/SE

**Table 6 — Predefined
Extended Polymarker
Bundle Table**

<i>Bundle Index</i>	<i>Marker Type</i>	<i>Marker Size Scale Factor</i>	<i>Colour Index†</i>
1	Dot	1.0	1
2	Plus	1.0	1
3	Asterisk	1.0	1
4	Circle	1.0	1
5	Cross	1.0	1
6	Square	1.0	1

† All Predefined Extended Polymarker Bundle entries (a SunPHIGS Extension based on PHIGS PLUS and not a part of the PHIGS standard) use colour model INDIRECT.

**Table 7 — Available
Text Fonts and
Precisions**

<i>Font Number</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Precisions Supported</i>
1	PFONT_MONO	PFONTMONO	All three Precisions (STRING, CHARACTER, and STROKE) are supported for all fonts
2	PFONT_SIMPLEX	PFONTSIMPLEX	
-1	PFONT_DUPLEX	PFONTDUPLEX	
-2	PFONT_COMPLEX	PFONTCOMPLEX	
-3	PFONT_TRIPLEX	PFONTTRIPLEX	
-4	PFONT_ITALIC_COMPLEX	PFONTITALICMPLX	
-5	PFONT_ITALIC_TRIPLEX	PFONTITALICTRPLX	
-6	PFONT_SCRIPT_SIMPLEX	PFONTSCRIPTSMPLX	
-7	PFONT_SCRIPT_COMPLEX	PFONTSCRIPTCMPLX	

**Table 8 — Predefined
Extended† Text
Bundle Table**

<i>Bundle Index</i>	<i>Font Number</i>	<i>Text Precision</i>	<i>Expansion Factor</i>	<i>Character Spacing</i>	<i>Colour Index†</i>
1	1 (Mono)	STROKE	1.0	0.0	1
2	2 (Simplex)	STROKE	1.0	0.0	1
3	-1 (Duplex)	STROKE	1.0	0.0	1
4	-2 (Complex)	STROKE	1.0	0.0	1
5	-3 (Triplex)	STROKE	1.0	0.0	1
6	-4 (Italic Complex)	STROKE	1.0	0.0	1
7	-5 (Italic Triplex)	STROKE	1.0	0.0	1
8	-6 (Script Simplex)	STROKE	1.0	0.0	1
9	-7 (Script Complex)	STROKE	1.0	0.0	1

† All Predefined Extended Text Bundle entries (a SunPHIGS Extension based on PHIGS PLUS and not a part of the PHIGS standard) use colour model INDIRECT.

Table 9 — Available Interior Styles

<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
PSTYLE_HOLLOW	PHOLLO	Hollow
PSTYLE_SOLID	PSOLID	Solid-filled
PSTYLE_HATCH	PHATCH	Hatched†
PSTYLE_EMPTY	PISEMP	Empty

† Hatched fill is not supported on TRUE COLOUR workstations.

Table 10 — Available Hatch Styles

<i>Style</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
-1	PHATCH_HORIZ	PHATHORZ	Horizontal Lines
-2	PHATCH_VERT	PHATVERT	Vertical Lines
-3	PHATCH_DIAG_45	PHATDIAG45	45-degree angled lines
-4	PHATCH_DIAG_135	PHATDIAG135	135-degree angled lines
-5	PHATCH_GRID_R	PHATGRID	Rectangular Grid
-6	PHATCH_GRID_D	PHATDIAGGRID	Diagonal Grid
-51	PHATCH_HORIZ_DBL	PHATHORZDB	Double-spaced Horizontal
-52	PHATCH_VERT_DBL	PHATVERTDB	Double-spaced Vertical
-53	PHATCH_DIAG_45_DBL	PHATDIAG45DB	Double-spaced 45 Degree
-54	PHATCH_DIAG_135_DBL	PHATDIAG135DB	Double-spaced 135 Degree
-55	PHATCH_GRID_R_DBL	PHATGRIDDB	Double-spaced Rectangular Grid
-56	PHATCH_GRID_D_DBL	PHATDIAGGRIDDB	Double-spaced Diagonal Grid
-101	PHATCH_HORIZ_TRANS	PHATHORZTR	Transparent Horizontal
-102	PHATCH_VERT_TRANS	PHATVERTTR	Transparent Vertical
-103	PHATCH_DIAG_45_TRANS	PHATDIAG45TR	Transparent 45 Degree
-104	PHATCH_DIAG_135_TRANS	PHATDIAG135TR	Transparent 135 Degree
-105	PHATCH_GRID_R_TRANS	PHATGRIDTR	Transparent Rectangular Grid
-106	PHATCH_GRID_D_TRANS	PHATDIAGGRIDTR	Transparent Diagonal Grid
-151	PHATCH_HORIZ_DBL_TRANS	PHATHORZTRDB	Double-spaced Transparent Horizontal
-152	PHATCH_VERT_DBL_TRANS	PHATVERTTRDB	Double-spaced Transparent Vertical

This table continues on the next page.

Table 10 — Available Hatch Styles

<i>Style</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
-153	PHATCH_DIAG_45_DBL_TRANS	PHATDIAG45TRDB	Double-spaced Transparent 45 Degree
-154	PHATCH_DIAG_135_DBL_TRANS	PHATDIAG135TRDB	Double-spaced Transparent 135 Degree
-155	PHATCH_GRID_R_DBL_TRANS	PHATGRIDTRDB	Double-spaced Transparent Rectangular
-156	PHATCH_GRID_D_DBL_TRANS	PHATDIAGGRIDTRDB	Double-spaced Transparent Diagonal

Table 11 — Predefined Fill Area Interior Bundle Table

<i>Bundle Index</i>	<i>Interior Style</i>	<i>Interior Style Index</i>	<i>Colour Index</i>	<i>Reflectance Equation†</i>	<i>Shading Method†</i>
1	Hollow	1	1	None	None
2	Solid	1	1	None	None
3	Hatch	-1 (Horizontal)	2	None	None
4	Hatch	-2 (Vertical)	3	None	None
5	Hatch	-3 (45 degree)	4	None	None

† This is a SunPHIGS Extension based on PHIGS PLUS and is not a part of the PHIGS standard. All Predefined Extended Interior Bundle entries use colour model INDIRECT; have back attribute values identical to the front; have the following area properties:

<i>Ambient Coefficient</i>	<i>Diffuse Coefficient</i>	<i>Specular Coefficient</i>	<i>Specular Colour</i>	<i>Specular Exponent</i>	<i>Transparency Coefficient</i>
1.0	1.0	1.0	(RGB, 1.0, 1.0, 1.0)	0.0	0.0

and have the following approximation data:

<i>Surface Approximation Type</i>	<i>Surface Approximation Values</i>	<i>Trimming Curve Approximation Type</i>	<i>Trimming Curve Approximation Value</i>
ws dependent	5.0, 5.0	ws dependent	5.0

**Table 12 —
Predefined Extended
Edge Bundle Table**

<i>Bundle Index</i>	<i>Edge Flag</i>	<i>Edgetype</i>	<i>Edge Width Scale Factor</i>	<i>Colour Index</i> †
1	ON	Solid	1.0	1
2	ON	Dashed	1.0	1
3	ON	Dotted	1.0	1
4	ON	Dash-dotted	1.0	1
5	ON	Long-dashed	1.0	1
6	ON	Dot-dashed-dot-dotted	1.0	1
7	ON	Center	1.0	1
8	ON	Phantom	1.0	1

† All Predefined Extended Edge Bundle entries (a SunPHIGS Extension based on PHIGS PLUS and not a part of the PHIGS standard) use colour model INDIRECT.

**Table 13 —
Predefined Colour
Table**

<i>Colour Index</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>	<i>Description</i>
0	0.0	0.0	0.0	Black
1	1.0	1.0	1.0	White
2	1.0	0.0	0.0	Red
3	0.0	1.0	0.0	Green
4	0.0	0.0	1.0	Blue
5	1.0	1.0	0.0	Yellow
6	0.0	1.0	1.0	Cyan
7	1.0	0.0	1.0	Magenta
≥8	1.0	1.0	1.0	White

**Table 14 — Available
Annotation Styles**

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
1	PANNO_STYLE_UNCONNECTED	PUNCON	Unconnected
2	PANNO_STYLE_LEAD_LINE	PLDLN	Lead Line

**Table 15 — Available
Generalized Drawing
Primitives 3**

<i>GDP Id</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
-2	PUGDP_CIRCLE3	PUGDPCIRC3	Circle 3
-3	PUGDP_CIRC_ARC3	PUGDPCIRCARC3	Circular Arc 3
-4	PUGDP_CIRC_ARC_CLOSE3	PUGDPCIRCARCCL3	Circular Arc Close 3
-5	PUGDP_ANNOT_CIRCLE3	PUGDPACIRC3	Annotation Circle 3
-6	PUGDP_ANNOT_CIRC_ARC3	PUGDPACIRCARC3	Annotation Circular Arc 3
-7	PUGDP_ANNOT_CIRC_ARC_CLOSE3	PUGDPACIRCARCCL3	Annotation Circular Arc Close 3
-8	PUGDP_ELLIPSE3	PUGDPELLP3	Ellipse 3
-9	PUGDP_ELLP_ARC3	PUGDPELLPARC3	Elliptical Arc 3
-10	PUGDP_ELLP_ARC_CLOSE3	PUGDPELLPARCCL3	Elliptical Arc Close 3
-11	PUGDP_ANNOT_ELLIPSE3	PUGDPAELLP3	Annotation Ellipse 3
-12	PUGDP_ANNOT_ELLP_ARC3	PUGDPAELLPARC3	Annotation Elliptical Arc 3
-13	PUGDP_ANNOT_ELLP_ARC_CLOSE3	PUGDPAELLPARCCL3	Annotation Elliptical Arc Close 3
-14	PUGDP_IMAGE3	PUGDPIMG3	Image 3
-15	PUGDP_RECT_GRID3	PUGDPRECTGRID3	Rectangular Grid 3
-16	PUGDP_RAD_GRID3	PUGDPRADGRID3	Radial Grid 3
-17	PUGDP_MONO_ENC_TXT3	PUGDPMONOENCTXT3	Mono- encoded Text 3
-18	PUGDP_MONO_ENC_ATXT_REL3	PUGDPENCATXTR3	Mono- encoded Annotation Text Relative 3
-19	PUGDP_HINTED_NURB_SURF3	PUGDPHINTEDNURBSURF3	Hinted NURBS Surface

**Table 16 — Available
Generalized Drawing
Primitives**

<i>GDP Id</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
-2	PUGDP_CIRCLE	PUGDPCIRC	Circle
-3	PUGDP_CIRC_ARC	PUGDPCIRCARC	Circular Arc
-4	PUGDP_CIRC_ARC_CLOSE	PUGDPCIRCARCCL	Circular Arc Close
-5	PUGDP_ANNOT_CIRCLE	PUGDPACIRC	Annotation Circle
-6	PUGDP_ANNOT_CIRC_ARC	PUGDPACIRCARC	Annotation Circular Arc
-7	PUGDP_ANNOT_CIRC_ARC_CLOSE	PUGDPACIRCARCCL	Annotation Circular Arc Close
-8	PUGDP_ELLIPSE	PUGDPELLP	Ellipse
-9	PUGDP_ELLP_ARC	PUGDPELLPARC	Elliptical Arc
-10	PUGDP_ELLP_ARC_CLOSE	PUGDPELLPARCCL	Elliptical Arc Close
-11	PUGDP_ANNOT_ELLIPSE	PUGDPAELLP	Annotation Ellipse
-12	PUGDP_ANNOT_ELLP_ARC	PUGDPAELLPARC	Annotation Elliptical Arc
-13	PUGDP_ANNOT_ELLP_ARC_CLOSE	PUGDPAELLPARCCL	Annotation Elliptical Arc Close
-14	PUGDP_IMAGE	PUGDPIMG	Image
-15	PUGDP_RECT_GRID	PUGDPRECTGRID	Rectangular Grid
-16	PUGDP_RAD_GRID	PUGDPRADGRID	Radial Grid
-17	PUGDP_MONO_ENC_TXT	PUGDPMONOENCTXT	Mono-encoded Text
-18	PUGDP_MONO_ENC_ATXT_REL	PUGDPMONOATXTR3	Mono-encoded Annotation Text Relative
-19‡	PUGDP_HINTED_NURBS_SURF3	PUGDPHINTEDNURBSURF3	Hinted NURBS Surface

‡ This primitive exists only as a GDP 3; there is no GDP equivalent.

Table 17 — Attributes used by Generalized Drawing Primitives and by Generalized Drawing Primitives 3

<i>GDP or GDP3 Id</i>	<i>Name</i>	<i>Attributes Used</i>
-2	Circle (3)	PATTR_INT, PATTR_EDGE
-3	Circular Arc (3)	PATTR_LINE
-4	Circular Arc Close (3)	PATTR_INT, PATTR_EDGE
-5	Annotation Circle (3)	PATTR_INT, PATTR_EDGE
-6	Annotation Circular Arc (3)	PATTR_LINE
-7	Annotation Circular Arc Close (3)	PATTR_INT, PATTR_EDGE
-8	Ellipse (3)	PATTR_INT, PATTR_EDGE
-9	Elliptical Arc (3)	PATTR_LINE
-10	Elliptical Arc Close (3)	PATTR_INT, PATTR_EDGE
-11	Annotation Ellipse (3)	PATTR_INT, PATTR_EDGE
-12	Annotation Elliptical Arc (3)	PATTR_LINE
-13	Annotation Elliptical Arc Close (3)	PATTR_INT, PATTR_EDGE
-14	Image (3)	NONE
-15	Rectangular Grid (3)	PATTR_INT, PATTR_EDGE
-16	Radial Grid (3)	PATTR_INT, PATTR_EDGE
-17	Mono-encoded Text (3)	PATTR_TEXT, PATTR_EDGE
-18	Mono-encoded Annotation Text Relative (3)	PATTR_TEXT, PATTR_EDGE
-19‡	Hinted NURBS Surface	PINATT, PEDATT

‡ This primitive exists only as a GDP 3; there is no GDP equivalent.

Table 18 — Polyline Shading Methods†

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
1	PSD_NONE	PSDNON	No Shading
2	PSD_COLOUR	PSDCOL	Colour Shading
3	PSD_DOT_PRODUCT	PSDDOT	Dot Product Shading
4	PSD_NORMAL	PSDNOR	Normal Shading

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

Table 19 — Available Interior Shading Methods†

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
1	PSD_NONE	PSDNON	No Shading
2	PSD_COLOUR	PSDCOL	Colour Shading

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

Table 20 — Available Light Source Types†

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
1	PLIGHT_AMBIENT	PAMB	Ambient Light Source
2	PLIGHT_DIRECTIONAL	PDIR	Directional Light Source
3	PLIGHT_POSITIONAL	PPOS	Positional Light Source
4	PLIGHT_SPOT	PSPOT	Spot Light Source

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

Table 21 — Available Reflectance Equations†

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
1	PREFL_NONE	PRNONE	No Reflectance Calculation Performed
2	PREFL_AMBIENT	PRAMB	Use Ambient Term
3	PREFL_AMB_DIFF	PRAMBD	Use Ambient and Diffuse Terms
4	PREFL_AMB_DIFF_SPEC	PRAMBDS	Use Ambient, Diffuse, and Specular Terms

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

Table 22 — Available Curve Approximation Methods†

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
1	PCURV_WS_DEP	PCAWSD	Workstation Dependent
2	PCURV_CONSTANT_PARAMETRIC_BETWEEN_KNOTS	PCAEPK	Constant Parametric Subdivision Between Knots
3	PCURV_CHORDAL_SIZE_WC	PCACSW	Chordal Size in WC
4	PCURV_CHORDAL_SIZE_NPC	PCACSN	Chordal Size in NPC
5	PCURV_CHORDAL_SIZE_DC	PCACSD	Chordal Size in DC
6	PCURV_CHORDAL_DEVIATION_WC	PCACDW	Chordal Deviation in WC
7	PCURV_CHORDAL_DEVIATION_NPC	PCACDN	Chordal Deviation in NPC
8	PCURV_CHORDAL_DEVIATION_DC	PCACDC	Chordal Deviation in DC

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

Table 23 — Available Surface Approximation Methods† ‡

<i>Value</i>	<i>C Name</i>	<i>FORTTRAN Name</i>	<i>Description</i>
1	PSURF_WS_DEP	PSAWSD	Workstation Dependent
2	PSURF_CONSTANT_PARAMETRIC_BETWEEN_KNOTS	PSAEPK	Constant Parametric Subdivision Between Knots
3	PSURF_CHORDAL_WC	PSACSW	Chordal Size in WC
4	PSURF_CHORDAL_NPC	PSACSN	Chordal Size in NPC
5	PSURF_CHORDAL_DC	PSACSD	Chordal Size in DC
6	PSURF_PLANAR_DEVIATION_WC	PSAPDW	Planar Deviation in WC
7	PSURF_PLANAR_DEVIATION_NPC	PSAPDN	Planar Deviation in NPC
8	PSURF_PLANAR_DEVIATION_DC	PSAPDD	Planar Deviation in DC

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

‡ In the current SunPHIGS release NURB surface parameter limits are used for trimmed surfaces: they control the placement of isoparametric curves when UNIFORM isoparametric curve placement is specified. In this case, the isoparametric curves are drawn only within the parameter limits and are spaced evenly between them.

**Table 24 —
Predefined Depth
Cue Table†**

<i>Depth Cue Index</i>	<i>Depth Cue Mode</i>	<i>Depth Cue Reference Planes</i>	<i>Depth Cue Scale Factors</i>	<i>Depth Cue Colour</i>
0	SUPPRESSED	(0.0,1.0)	(1.0,1.0)	(INDIRECT,0)
1	ALLOWED	(0.0,1.0)	(0.0,1.0)	(INDIRECT,0)

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

**Table 25 —
Predefined Light
Sources†**

<i>Index</i>	<i>Type</i>	<i>Data Record</i>
1	AMBIENT	(INDIRECT,1)
2	DIRECTIONAL	(INDIRECT,1),1.0,1.0,1.0
3	DIRECTIONAL	(INDIRECT,1),-1.0,1.0,1.0
4	DIRECTIONAL	(INDIRECT,1),1.0,-1.0,1.0
5	DIRECTIONAL	(INDIRECT,1),1.0,1.0,-1.0

† These are SunPHIGS Extensions based on PHIGS PLUS and are not part of the PHIGS standard.

**Table 26 —
Maximum Display
Space Size†**

<i>Workstation Type</i>	<i>Initial Device Coordinates</i>	<i>Initial Device Address Units</i>
X Tool	(600.0, 600.0, 1.0)	(600, 600, 1)
X Drawable	(Drawable Width, Drawable Height, 1.0)	(Drawable Width, Drawable Height, 1)
CGM Output	(600.0, 600.0, 1.0)	(32767, 32767, 1)
X Drawable Region	(Region Width, Region Height, 1.0)	(Region Width, Region Height, 1)

**Table 27 — Available
Polyline, Text, and
Edge End Caps†**

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
0	PHIGS_END_BUTT	PHIGSENDBUTT	Butt
1	PHIGS_END_SQUARE	PHIGSEND SQUARE	Squared
2	PHIGS_END_ROUND	PHIGSENDROUND	Round

† Non-GT workstations only support the BUTT end cap. GT workstations support all three end caps.

**Table 28 — Available
Polyline, Text, and
Edge Join Styles†**

<i>Value</i>	<i>C Name</i>	<i>FORTRAN Name</i>	<i>Description</i>
0	PHIGS_JOIN_BUTT	PHIGSJOINBUTT	Butt
1	PHIGS_JOIN_BEVELED	PHIGSJOINBEVELED	Beveled
2	PHIGS_JOIN_MITRED	PHIGSJOINMITRED	Mitred
3	PHIGS_JOIN_ROUND	PHIGSJOINROUND	Round

† Non-GT workstations only support the BUTT join style. GT workstations support all four join styles.

SEE ALSO

INQUIRE WORKSTATION CONNECTION AND TYPE (3P)
 PHIGS DESCRIPTION TABLE (7P)
 WORKSTATION TYPE CREATE (3P)
 WORKSTATION TYPE GET (3P)
 WORKSTATION TYPE SET (3P)

Index

C

CGM, 3
Character Sets
 INTRO INTERNATIONALIZATION, 21
COLOUR, 9
Colour
 COLOUR, 9

D

DESCRIPTION TABLE, 27
Dynamic Modifications
 PHIGS WORKSTATION DESCRIPTION
 TABLE, 35

E

Elements, attributes, see *Attributes*
Elements, primitives, see *Primitives*
Elements, transformations, see *Transformation Matrix*

G

Generalized Drawing Primitives
 PHIGS WORKSTATION DESCRIPTION
 TABLE, 35
Generalized Drawing Primitives 3
 PHIGS WORKSTATION DESCRIPTION
 TABLE, 35
Generalized Structure Elements
 PHIGS WORKSTATION DESCRIPTION

TABLE,

Generalized Structure Elements, *continued*
35

I

INTRO DRAWABLE REGION, 19
INTRO INTERNATIONALIZATION, 21
Introduction To SunPHIGS Tables and Explanations,
1

P

PHIGS DESCRIPTION TABLE, 27
PHIGS TRAVERSAL STATE LIST, 29
PHIGS WORKSTATION DESCRIPTION TABLE, 35

T

Tables and Explanations, description, 1
TRAVERSAL STATE LIST, 29

W

Workstation Description Table
 PHIGS WORKSTATION DESCRIPTION
 TABLE, 35
Workstation Types
 PHIGS WORKSTATION DESCRIPTION
 TABLE, 35