STD.ISO 9593-4-ENGL 1991 ■ 4851903 0782231 212 ■

INTERNATIONAL STANDARD

ISO/IEC 9593-4

First edition
1991-12-15

AMENDMENT 2
1998-12-15

Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 4:
C

AMENDMENT 2:
Incorporation of PHIGS amendments

Technologies de l'information — Infographie — Interfaces langage avec PHIGS —

Partie 4: C

AMENDEMENT 2: Incorporation des amendements de PHIGS

This material is reproduced from ISO documents under International Organization for Standardization (ISO) Copyright License Number IHS/ICC/1996. Not for resale. No part of these ISO documents may be reproduced in any form, electronic retrieval system or otherwise, except as allowed in the copyright law of the country of use, or with the prior written consent of ISO (Case postale 56, 1211 Geneva 20, Switzerland, Fax +41 22 734 10 79), IHS or the ISO Licensor's members.

Reference number
ISO/IEC 9593-4:1991/Amd.2:1998 (E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 2 to ISO/IEC 9593-4:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics and image processing*.

© ISO/IEC 1998

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

© ISO/IEC

Information technology - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings -

Part 4:
C

Amendment 2: Incorporation of PHIGS amendments

Page 2

2 Normative references

The number of the first reference is changed to ISO/IEC 9592-1:1996.

Page 3

3 The C language binding of PHIGS

3.1 Conformance

The following text replaces the content of 3.1:

"This binding incorporates the rules of conformance defined in the PHIGS Standard (ISO/IEC 9592) for PHIGS implementations, with those additional requirements specifically defined for C language implementations of PHIGS.

The following criteria are established for determining conformance of an implementation to this binding:

a) In order to conform to the *Basic PHIGS profile*, an implementation of the C binding of PHIGS shall implement at least the functionality defined for the *Basic PHIGS profile* as specified in ISO/IEC

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9592-1. It shall make visible all of the declarations in the C binding specified in clauses 5, 6, and 7 of this part of ISO/IEC 9593.

b) In order to conform to the *PHIGS PLUS profile*, an implementation of the C binding of PHIGS shall implement at least the functionality defined for the *PHIGS PLUS profile* as specified in ISO/IEC 9592-1. It shall make visible all of the declarations in the C binding specified in clauses 5 through 10 of this part of ISO/IEC 9593.

c) In order to conform to the *Full PHIGS profile*, an implementation of the C binding of PHIGS shall implement all of the functionality defined for the *Full PHIGS profile* as specifed in ISO/IEC 9592-1. It shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 9593.

d) The syntax of the function names shall be precisely as specified in this part of ISO/IEC 9593 and the parameters shall be of the data types stated in this part of ISO/IEC 9593.

Pages 10 to 13

4 Tables

4.2 Table of abbreviations

The following abbreviations are added alphabetically to Table 1

"

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
acknowledgement	ack
address	addr
associate	assoc
association	assoc
attribute	attr
automatic	auto
background	backg
boundary	bound
buffer	buf
centre	ctr
channel	chan
complete	comp
composition	compos
conditions	conds
configuration	config
continuity	cont
correlate, correlation	corr
define	def
destination	dest
disassociate	disassoc
disposition	dispos

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 1 - Abbreviations ordered alphabetically (Continued)

Word or Phrase	Abbreviation
distance	dist
frequency	freq
group	grp
groups	grps
heuristics	heur
include	incl
incompatible	incompat
incomplete	incomp
indicator	indic
instance	inst
logical input device	lid
manipulation	manip
measure	meas
non-atomic	na
non-retained	nr
non-retained data	nrd
optimization	opt
parameterization	param
parameters	params
picture	pict
posting	post
process, processing	proc
reference	ref
registered	reg
render, rendering	rend
resource	res
sampling	sampl
source	src
target	targ
texture	textr
transfer	trans
traverse, traversal	trav
trigger	trig
uncompressed	uncomp
undefine	undef
weight	wt
which	NULL

"

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pages 14 to 20

4.3 Function names

4.3.1 List ordered alphabetically by bound name

The following function names are merged alphabetically by bound name in Table 2

"

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
passoc_image_res (...)	ASSOCIATE IMAGE RESOURCE
passoc_trav_res (...)	ASSOCIATE TRAVERSAL RESOURCE
pattach_lid_to_light_src (...)	ATTACH LOGICAL INPUT DEVICE TO LIGHT SOURCE
pattach_lid_to_view (...)	ATTACH LOGICAL INPUT DEVICE TO VIEW
pcircle (...)	CIRCLE
pcircle3 (...)	CIRCLE 3
pcircular_arc (...)	CIRCULAR ARC
pcircular_arc3 (...)	CIRCULAR ARC 3
pcircular_arc_close (...)	CIRCULAR ARC CLOSE
pcircular_arc_close3 (...)	CIRCULAR ARC CLOSE 3
pclear_targ (...)	CLEAR TARGET
pclose_di_struct (...)	CLOSE DIRECT INTERPRETATION STRUCTURE
pcond_exec_struct (...)	CONDITIONAL EXECUTE STRUCTURE
pcond_inst_struct (...)	CONDITIONAL INSTANCE STRUCTURE
pcond_return (...)	CONDITIONAL RETURN
pcond_skip_elements (...)	CONDITIONAL SKIP ELEMENTS
pcond_skip_to_label (...)	CONDITIONAL SKIP TO LABEL
pcopy_elem_struct (...)	COPY ELEMENT FROM STRUCTURE
pcopy_elem_range_struct (...)	COPY ELEMENT RANGE FROM STRUCTURE
pcopy_elems_between_labels_struct (...)	COPY ELEMENTS BETWEEN LABELS FROM STRUCTURE
pcopy_targ (...)	COPY TARGET
pcreate_composite_measure (...)	CREATE COMPOSITE MEASURE
pcreate_mipmap_texture (...)	CREATE MIPMAP TEXTURE
pcreate_set_measure (...)	CREATE SET MEASURE
pcreate_targ (...)	CREATE TARGET
pdefine_choice (...)	DEFINE CHOICE
pdefine_composite (...)	DEFINE COMPOSITE
pdefine_linetype (...)	DEFINE LINETYPE
pdefine_locator (...)	DEFINE LOCATOR
pdefine_marker_type (...)	DEFINE MARKER TYPE
pdefine_pick (...)	DEFINE PICK
pdefine_post_grp (...)	DEFINE POSTING GROUP
pdefine_set (...)	DEFINE SET
pdefine_string (...)	DEFINE STRING
pdefine_stroke (...)	DEFINE STROKE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pdefine_valuator (...)	DEFINE VALUATOR
pdestroy_composite_measure (...)	DESTROY COMPOSITE MEASURE
pdestroy_set_measure (...)	DESTROY SET MEASURE
pdestroy_targ (...)	DESTROY TARGET
pdetach_lid_from_light_src (...)	DETACH LOGICAL INPUT DEVICE FROM LIGHT SOURCE
pdetach_lid_from_view (...)	DETACH LOGICAL INPUT DEVICE FROM VIEW
pdisable_di_pick (...)	DISABLE DIRECT INTERPRETATION PICK
pdisassoc_image_res (...)	DISASSOCIATE IMAGE RESOURCE
pdisassoc_trav_res (...)	DISASSOCIATE TRAVERSAL RESOURCE
pellipse (...)	ELLIPSE
pellipse3 (...)	ELLIPSE 3
pelliptical_arc (...)	ELLIPTICAL ARC
pelliptical_arc3 (...)	ELLIPTICAL ARC 3
pelliptical_arc_close (...)	ELLIPTICAL ARC CLOSE
pelliptical_arc_close3 (...)	ELLIPTICAL ARC CLOSE 3
penable_di_pick (...)	ENABLE DIRECT INTERPRETATION PICK
pend_watch_on_elem_range (...)	END WATCH ON ELEMENT RANGE
pfill_circle (...)	FILL CIRCLE
pfill_circle3 (...)	FILL CIRCLE 3
pfill_ellipse (...)	FILL ELLIPSE
pfill_ellipse3 (...)	FILL ELLIPSE 3
pget_composite (...)	GET COMPOSITE
pget_composite3 (...)	GET COMPOSITE 3
pget_set (...)	GET SET
pget_set3 (...)	GET SET 3
pinit_composite (...)	INITIALIZE COMPOSITE
pinit_composite3 (...)	INITIALIZE COMPOSITE 3
pinit_di_pick (...)	INITIALIZE DIRECT INTERPRETATION PICK
pinit_di_pick3 (...)	INITIALIZE DIRECT INTERPRETATION PICK 3
pinit_set (...)	INITIALIZE SET
pinit_set3 (...)	INITIALIZE SET 3
pinq_alpha_facs (...)	INQUIRE ALPHA FACILITIES
pinq_appl_filter (...)	INQUIRE APPLICATION FILTER
pinq_assoc_image_res (...)	INQUIRE ASSOCIATED IMAGE RESOURCES
pinq_back_clip_indicator (...)	INQUIRE BACK CLIPPING INDICATOR
pinq_back_plane_dist (...)	INQUIRE BACK PLANE DISTANCE
pinq_choice_facs (...)	INQUIRE CHOICE FACILITIES
pinq_composite_facs (...)	INQUIRE COMPOSITE FACILITIES
pinq_composite_meas_def (...)	INQUIRE COMPOSITE MEASURE DEFINITION
pinq_composite_st (...)	INQUIRE COMPOSITE DEVICE STATE
pinq_composite_st3 (...)	INQUIRE COMPOSITE DEVICE STATE 3
pinq_cond_trav_facs (...)	INQUIRE CONDITIONAL TRAVERSAL FACILITIES
pinq_config_setting_facs (...)	INQUIRE CONFIGURATION SETTING FACILITIES
pinq_dc_clip_regions (...)	INQUIRE DEVICE COORDINATE CLIP REGIONS
pinq_dc_clip_regions3 (...)	INQUIRE DEVICE COORDINATE CLIP REGIONS 3

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pinq_dc_clip_regions_facs(...)	INQUIRE DEVICE COORDINATE CLIP REGIONS FACILITIES
pinq_def_composite_data (...)	INQUIRE DEFAULT COMPOSITE DEVICE DATA
pinq_def_composite_data3 (...)	INQUIRE DEFAULT COMPOSITE DEVICE DATA 3
pinq_def_di_pick_data (...)	INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA
pinq_def_di_pick_data3 (...)	INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 3
pinq_def_set_data (...)	INQUIRE DEFAULT SET DEVICE DATA
pinq_def_set_data3 (...)	INQUIRE DEFAULT SET DEVICE DATA 3
pinq_def_targ_disp (...)	INQUIRE DEFAULT TARGET DISPOSITION
pinq_di_mode (...)	INQUIRE DIRECT INTERPRETATION MODE
pinq_di_pick_corr_point (...)	INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT
pinq_di_pick_corr_point3 (...)	INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 3
pinq_di_pick_set_status (...)	INQUIRE DIRECT INTERPRETATION PICK SET STATUS
pinq_di_pick_st (...)	INQUIRE DIRECT INTERPRETATION PICK STATE
pinq_di_pick_st3 (...)	INQUIRE DIRECT INTERPRETATION PICK STATE 3
pinq_disp_targ (...)	INQUIRE DISPLAY TARGET
pinq_di_trav_facs (...)	INQUIRE DIRECT INTERPRETATION TRAVERSAL FACILITIES
pinq_dyns_post_grps (...)	INQUIRE DYNAMICS OF POSTING GROUPS
pinq_dyns_ws_attrs_texture (...)	INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE
pinq_edge_rep_full (...)	INQUIRE EDGE REPRESENTATION FULL
pinq_ext_pat_facs (...)	INQUIRE EXTENDED PATTERN FACILITIES
pinq_ext_pat_rep (...)	INQUIRE EXTENDED PATTERN REPRESENTATION
pinq_front_clip_indicator (...)	INQUIRE FRONT CLIPPING INDICATOR
pinq_front_plane_dist (...)	INQUIRE FRONT PLANE DISTANCE
pinq_grps_posted (...)	INQUIRE SET OF GROUPS TO WHICH POSTED
pinq_highl_facs (...)	INQUIRE HIGHLIGHTING FACILITIES
pinq_highl_rep (...)	INQUIRE HIGHLIGHTING REPRESENTATION
pinq_image_res (...)	INQUIRE IMAGE RESOURCE
pinq_image_res_facs (...)	INQUIRE IMAGE RESOURCE FACILITIES
pinq_lid_attached_to_light_src (...)	INQUIRE LOGICAL INPUT DEVICE ATTACHED TO LIGHT SOURCE
pinq_lid_attached_to_view (...)	INQUIRE LOGICAL INPUT DEVICE ATTACHED TO VIEW
pinq_lid_def (...)	INQUIRE LOGICAL INPUT DEVICE DEFINITION
pinq_line_rep_full (...)	INQUIRE POLYLINE REPRESENTATION FULL
pinq_linetype_def (...)	INQUIRE LINETYPE DEFINITION
pinq_linetype_def_facs (...)	INQUIRE LINETYPE DEFINITION FACILITIES
pinq_linetype_def_support (...)	INQUIRE LINETYPE DEFINITION SUPPORT
pinq_list_def_appl_filters (...)	INQUIRE LIST OF DEFINED APPLICATION FILTERS
pinq_list_highl_inds (...)	INQUIRE LIST OF HIGHLIGHTING INDICES

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pinq_list_post_grps (...)	INQUIRE POSTED STRUCTURES FROM POSTING GROUP
pinq_list_texture_inds (...)	INQUIRE LIST OF TEXTURE INDICES
pinq_lists_avail_lids (...)	INQUIRE LISTS OF AVAILABLE LOGICAL INPUT DEVICES
pinq_loc_facs (...)	INQUIRE LOCATOR FACILITIES
pinq_marker_type_def (...)	INQUIRE MARKER TYPE DEFINITION
pinq_marker_type_def_facs (...)	INQUIRE MARKER TYPE DEFINITION FACILITIES
pinq_marker_type_def_support (...)	INQUIRE MARKER TYPE DEFINITION SUPPORT
pinq_mipmap_facs (...)	INQUIRE MIPMAP FACILITIES
pinq_num_avail_na_in (...)	INQUIRE NUMBER OF AVAILABLE NON-ATOMIC LOGICAL INPUT DEVICES
pinq_num_def_linetypes (...)	INQUIRE NUMBER OF DEFINED LINETYPES
pinq_num_def_marker_types (...)	INQUIRE NUMBER OF DEFINED MARKER TYPES
pinq_num_pred_appl_filters (...)	INQUIRE NUMBER OF PREDEFINED APPLICATION FILTERS
pinq_pick_facs (...)	INQUIRE PICK FACILITIES
pinq_pick_mapping_facs (...)	INQUIRE PICK MAPPING FACILITIES
pinq_pick_mapping_state (...)	INQUIRE PICK MAPPING STATE
pinq_pict_status (...)	INQUIRE PICTURE STATUS
pinq_posted_di_struct (...)	INQUIRE POSTED DIRECT INTERPRETATION STRUCTURE
pinq_posted_structs_from_post_grp (...)	INQUIRE POSTED STRUCTURES FROM POSTING GROUP
pinq_post_grp (...)	INQUIRE POSTING GROUP
pinq_post_grp_facs (...)	INQUIRE POSTING GROUP FACILITIES
pinq_pred_appl_filter (...)	INQUIRE PREDEFINED APPLICATION FILTER
pinq_pred_assoc_targ_trav_res (...)	INQUIRE PREDEFINED ASSOCIATION OF TARGET WITH TRAVERSAL RESOURCES
pinq_pred_assoc_trav_res_targ (...)	INQUIRE PREDEFINED ASSOCIATION OF TRAVERSAL RESOURCE WITH TARGET
pinq_pred_composite_meas_def (...)	INQUIRE PREDEFINED COMPOSITE MEASURE DEFINITION
pinq_pred_edge_rep_full (...)	INQUIRE PREDEFINED EDGE REPRESENTATION FULL
pinq_pred_ext_pat_rep (...)	INQUIRE PREDEFINED EXTENDED PATTERN REPRESENTATION
pinq_pred_highl_rep (...)	INQUIRE PREDEFINED HIGHLIGHTING REPRESENTATION
pinq_pred_image_res (...)	INQUIRE PREDEFINED IMAGE RESOURCE
pinq_pred_line_rep_full (...)	INQUIRE PREDEFINED POLYLINE REPRESENTATION FULL
pinq_pred_linetype_def (...)	INQUIRE PREDEFINED LINETYPE DEFINITION
pinq_pred_marker_type_def (...)	INQUIRE PREDEFINED MARKER TYPE DEFINITION
pinq_pred_post_grp (...)	INQUIRE PREDEFINED POSTING GROUP
pinq_pred_set_meas_def (...)	INQUIRE PREDEFINED SET MEASURE DEFINITION
pinq_pred_texture_rep (...)	INQUIRE PREDEFINED TEXTURE REPRESENTATION

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pinq_proj_ref_point (...)	INQUIRE PROJECTION REFERENCE POINT
pinq_proj_type (...)	INQUIRE PROJECTION TYPE
pinq_proj_vp (...)	INQUIRE PROJECTION VIEWPORT
pinq_proj_vp3 (...)	INQUIRE PROJECTION VIEWPORT 3
pinq_rend_targ (...)	INQUIRE RENDERING TARGET
pinq_set_facs (...)	INQUIRE SET FACILITIES
pinq_set_meas_def (...)	INQUIRE SET MEASURE DEFINITION
pinq_set_st (...)	INQUIRE SET DEVICE STATE
pinq_set_st3 (...)	INQUIRE SET DEVICE STATE 3
pinq_string_facs (...)	INQUIRE STRING FACILITIES
pinq_stroke_facs (...)	INQUIRE STROKE FACILITIES
pinq_targ_assoc_trav_res (...)	INQUIRE TARGET ASSOCIATED WITH TRAVERSAL RESOURCE
pinq_targ_dispos (...)	INQUIRE TARGET DISPOSITION
pinq_targ_facs (...)	INQUIRE TARGET FACILITIES
pinq_targ_manip_mode (...)	INQUIRE TARGET MANIPULATION MODE
pinq_targ_st (...)	INQUIRE TARGET STATE
pinq_texture_facs (...)	INQUIRE TEXTURE FACILITIES
pinq_texture_map_facs (...)	INQUIRE TEXTURE MAPPING FACILITIES
pinq_texture_rep (...)	INQUIRE TEXTURE REPRESENTATION
pinq_transparency_mode (...)	INQUIRE TRANSPARENCY MODE
pinq_transparency_thresholds (...)	INQUIRE TRANSPARENCY THRESHOLDS
pinq_trav_res_assoc_targ (...)	INQUIRE TRAVERSAL RESOURCES ASSOCIATED WITH TARGET
pinq_trav_res_facs (...)	INQUIRE TRAVERSAL RESOURCE FACILITIES
pinq_val_facs (...)	INQUIRE VALUATOR FACILITIES
pinq_view_plane_dist (...)	INQUIRE VIEW PLANE DISTANCE
pinq_view_plane_norm (...)	INQUIRE VIEW PLANE NORMAL
pinq_view_ref_point (...)	INQUIRE VIEW REFERENCE POINT
pinq_view_ref_point3 (...)	INQUIRE VIEW REFERENCE POINT 3
pinq_view_status (...)	INQUIRE VIEW STATUS
pinq_view_up_vec (...)	INQUIRE VIEW UP VECTOR
pinq_view_up_vec3 (...)	INQUIRE VIEW UP VECTOR 3
pinq_view_win_limits (...)	INQUIRE VIEW WINDOW LIMITS
pinq_watch_elem_range (...)	INQUIRE WATCH ON ELEMENT RANGE
pinq_watch_range_st (...)	INQUIRE WATCH RANGE STATE
pinq_wss_di_struct_posted (...)	INQUIRE SET OF WORKSTATIONS TO WHICH DIRECT INTERPRETATION STRUCTURE POSTED
pinq_ws_st_table_highl (...)	INQUIRE WORKSTATION STATE TABLE LENGTHS HIGHLIGHTING
pinq_ws_st_table_texture (...)	INQUIRE WORKSTATION STATE TABLE LENGTHS TEXTURE
pinq_xy_clip_indicator (...)	INQUIRE X-Y CLIPPING INDICATOR
pinst_struct (...)	INSTANCE STRUCTURE
pmanip_trav_res (...)	MANIPULATE TRAVERSAL RESOURCE
pmap_dc_point_to_pick_paths (...)	MAP DEVICE COORDINATE POINT TO PICK PATHS

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pmap_dc_to_wc (...)	MAP DEVICE COORDINATES TO WORLD COORDINATES
pmap_dc_to_wsc (...)	MAP DEVICE COORDINATES TO WINDOW SYSTEM COORDINATES
pmap_wsc_to_dc (...)	MAP WINDOW SYSTEM COORDINATES TO DEVICE COORDINATES
pmark_multi_pass_compl (...)	MARK MULTI-PASS COMPLETION
pmark_multi_pass_start (...)	MARK MULTI-PASS START
pmark_pass_compl (...)	MARK PASS COMPLETION
pmark_pass_start (...)	MARK PASS START
pmove_elem_struct (...)	MOVE ELEMENT FROM STRUCTURE
pmove_elem_range_struct (...)	MOVE ELEMENT RANGE FROM STRUCTURE
pmove_elems_between_labels_struct (...)	MOVE ELEMENTS BETWEEN LABELS FROM STRUCTURE
popen_di_struct (...)	OPEN DIRECT INTERPRETATION STRUCTURE
ppop_st (...)	POP STATE
ppost_di_struct (...)	POST DIRECT INTERPRETATION STRUCTURE
ppost_struct_to_grp (...)	POST STRUCTURE TO GROUP
ppush_st (...)	PUSH STATE
predraw_all_from_grp_on_targ (	REDRAW ALL STRUCTURES FROM POSTING GROUP ON TARGET
predraw_all_structs_from_grp (...)	REDRAW ALL STRUCTURES FROM POSTING GROUP
predraw_all_structs_on_targ (...)	REDRAW ALL STRUCTURES ON TARGET
prenew_di_state (...)	RENEW DIRECT INTERPRETATION STATE
preq_composite (...)	REQUEST COMPOSITE
preq_composite3 (...)	REQUEST COMPOSITE 3
preq_set (...)	REQUEST SET
preq_set3 (...)	REQUEST SET 3
preset_all_trav_res (...)	RESET ALL TRAVERSAL RESOURCES
pret_num_passes_req (...)	RETRIEVE NUMBER OF PASSES REQUIRED
pret_window_system_colr (...)	RETRIEVE WINDOW SYSTEM COLOUR
psample_composite (...)	SAMPLE COMPOSITE
psample_composite3 (...)	SAMPLE COMPOSITE 3
psample_set (...)	SAMPLE SET
psample_set3 (...)	SAMPLE SET 3
pset_active_textures (...)	SET ACTIVE TEXTURES
pset_alpha_data_sel_ind (...)	SET ALPHA DATA SELECTION INDEX
pset_alpha_src_sel (...)	SET ALPHA SOURCE SELECTOR
pset_appl_filter (...)	SET APPLICATION FILTER
pset_appl_int (...)	SET APPLICATION INTEGER
pset_appl_real (...)	SET APPLICATION REAL
pset_back_active_textures (...)	SET BACK ACTIVE TEXTURES
pset_back_clip_indicator (...)	SET BACK CLIPPING INDICATOR
pset_back_plane_dist (...)	SET BACK PLANE DISTANCE
pset_back_transparency (...)	SET BACK TRANSPARENCY
pset_composite_mode (...)	SET COMPOSITE MODE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pset_composite_pick_filter (...)	SET COMPOSITE PICK FILTER
pset_cond_flags (...)	SET CONDITION FLAGS
pset_cond_flags_from_tests (...)	SET CONDITION FLAGS FROM TESTS
pset_dc_clip_regions (...)	SET DEVICE COORDINATE CLIP REGIONS
pset_dc_clip_regions3 (...)	SET DEVICE COORDINATE CLIP REGIONS 3
pset_depth_cue_rep_mask (	SET DEPTH CUE REPRESENTATION MASK
pset_di_mode (...)	SET DIRECT INTERPRETATION MODE
pset_di_pick_corr_point (...)	SET DIRECT INTERPRETATION PICK CORRELATION POINT
pset_di_pick_corr_point3 (...)	SET DIRECT INTERPRETATION PICK CORRELATION POINT 3
pset_di_pick_filter (...)	SET DIRECT INTERPRETATION PICK FILTER
pset_disp_targ (...)	SET DISPLAY TARGET
pset_edgecap (...)	SET EDGECAP
pset_edgejoin (...)	SET EDGEJOIN
pset_edgemitre_limit (...)	SET EDGEMITRE LIMIT
pset_edge_rep_full (...)	SET EDGE REPRESENTATION FULL
pset_edge_rep_mask (...)	SET EDGE REPRESENTATION MASK
pset_edgetype_adapt (...)	SET EDGETYPE ADAPTABILITY
pset_edgetype_cont (...)	SET EDGETYPE CONTINUITY
pset_edgetype_offset (...)	SET EDGETYPE OFFSET
pset_ext_pat_rep (...)	SET EXTENDED PATTERN REPRESENTATION
pset_front_clip_indicator (...)	SET FRONT CLIPPING INDICATOR
pset_front_plane_dist (...)	SET FRONT PLANE DISTANCE
pset_highl_ind (...)	SET HIGHLIGHTING INDEX
pset_highl_method (...)	SET HIGHLIGHTING METHOD
pset_highl_rep (...)	SET HIGHLIGHTING REPRESENTATION
pset_int_rep_mask (...)	SET INTERIOR REPRESENTATION MASK
pset_linecap (...)	SET LINECAP
pset_linejoin (...)	SET LINEJOIN
pset_linemitre_limit (...)	SET LINEMITRE LIMIT
pset_line_rep_full (...)	SET POLYLINE REPRESENTATION FULL
pset_line_rep_mask (...)	SET POLYLINE REPRESENTATION MASK
pset_linetype_adapt (...)	SET LINETYPE ADAPTABILITY
pset_linetype_cont (...)	SET LINETYPE CONTINUITY
pset_linetype_offset (...)	SET LINETYPE OFFSET
pset_marker_rep_mask (...)	SET POLYMARKER REPRESENTATION MASK
pset_pat_rep_mask (...)	SET PATTERN REPRESENTATION MASK
pset_pick_mapping_data (...)	SET PICK MAPPING DATA
pset_post_grp_backg_method (...)	SET POSTING GROUP BACKGROUND METHOD
pset_post_grp_backg_style (...)	SET POSTING GROUP BACKGROUND STYLE
pset_post_grp_border_ind (...)	SET POSTING GROUP BORDER INDEX
pset_post_grp_border_indicator (...)	SET POSTING GROUP BORDER INDICATOR
pset_post_grp_priority (...)	SET POSTING GROUP PRIORITY
pset_post_grp_status (...)	SET POSTING GROUP STATUS
pset_proj_ref_point (...)	SET PROJECTION REFERENCE POINT
pset_proj_type (...)	SET PROJECTION TYPE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
pset_proj_vp (...)	SET PROJECTION VIEWPORT
pset_proj_vp3 (...)	SET PROJECTION VIEWPORT 3
pset_refl_rep_mask (...)	SET REFLECTANCE REPRESENTATION MASK
pset_rend_targ (...)	SET RENDERING TARGET
pset_set_mode (...)	SET SET MODE
pset_set_pick_filter (...)	SET SET PICK FILTER
pset_st_visual_rep (...)	SET STATE OF VISUAL REPRESENTATION
pset_targ_dispos (...)	SET TARGET DISPOSITION
pset_targ_manip_mode (...)	SET TARGET MANIPULATION MODE
pset_targ_st_visual_rep (...)	SET TARGET STATE OF VISUAL REPRESENTATION
pset_text_rep_mask (...)	SET TEXT REPRESENTATION MASK
pset_texture_binding (...)	SET TEXTURE BINDING
pset_texture_composition (...)	SET TEXTURE COMPOSITION
pset_texture_param (...)	SET TEXTURE PARAMETRIZATION
pset_texture_perspect_corr (...)	SET TEXTURE PERSPECTIVE CORRECTION
pset_texture_rep (...)	SET TEXTURE REPRESENTATION
pset_texture_rep_mask (...)	SET TEXTURE REPRESENTATION MASK
pset_texture_res_opt_heur (...)	SET TEXTURE RESOURCE OPTIMIZATION HEURISTICS
pset_texture_sampling (...)	SET TEXTURE SAMPLING
pset_texture_sampling_freq (...)	SET TEXTURE SAMPLING FREQUENCY
pset_transparency (...)	SET TRANSPARENCY
pset_transparency_mode (...)	SET TRANSPARENCY MODE
pset_transparency_thresholds (...)	SET TRANSPARENCY THRESHOLDS
pset_view_plane_dist (...)	SET VIEW PLANE DISTANCE
pset_view_plane_norm (...)	SET VIEW PLANE NORMAL
pset_view_ref_point (...)	SET VIEW REFERENCE POINT
pset_view_ref_point3 (...)	SET VIEW REFERENCE POINT 3
pset_view_up_vec (...)	SET VIEW UP VECTOR
pset_view_up_vec3 (...)	SET VIEW UP VECTOR 3
pset_view_win_limits (...)	SET VIEW WINDOW LIMITS
pset_watch_on_elem_range (...)	SET WATCH ON ELEMENT RANGE
pset_xy_clip_indicator (...)	SET X-Y CLIPPING INDICATOR
ptrans_di_pick_set (...)	TRANSFER DIRECT INTERPRETATION PICK SET
pundefine_choice (...)	UNDEFINE CHOICE
pundefine_composite (...)	UNDEFINE COMPOSITE
pundefine_locator (...)	UNDEFINE LOCATOR
pundefine_pick (...)	UNDEFINE PICK
pundefine_post_grp (...)	UNDEFINE POSTING GROUP
pundefine_set (...)	UNDEFINE SET
pundefine_string (...)	UNDEFINE STRING
pundefine_stroke (...)	UNDEFINE STROKE
pundefine_valuator (...)	UNDEFINE VALUATOR
punpost_all_structs_from_grp (...)	UNPOST ALL STRUCTURES FROM GROUP
punpost_di_struct (...)	UNPOST DIRECT INTERPRETATION STRUCTURE
punpost_structs_from_grps (...)	UNPOST STRUCTURES FROM GROUPS

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name (Continued)

C Name	PHIGS and PHIGS PLUS Name
`pupd_targ (...)`	UPDATE TARGET
`pupd_view_rep (...)`	UPDATE VIEW REPRESENTATION
`pws_type_create (...)`	WORKSTATION TYPE CREATE
`pws_type_destroy (...)`	WORKSTATION TYPE DESTROY
`pws_type_get (...)`	WORKSTATION TYPE GET
`pws_type_set (...)`	WORKSTATION TYPE SET

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

”

Pages 20 to 26

4.3.2 List ordered alphabetically by PHIGS and PHIGS PLUS name

The following function names are merged alphabetically by function name in Table 3:

“

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
ASSOCIATE IMAGE RESOURCE	`passoc_image_res (...)`
ASSOCIATE TRAVERSAL RESOURCE	`passoc_trav_res (...)`
ATTACH LOGICAL INPUT DEVICE TO LIGHT SOURCE	`pattach_lid_to_light_src (...)`
ATTACH LOGICAL INPUT DEVICE TO VIEW	`pattach_lid_to_view (...)`
CIRCLE	`pcircle (...)`
CIRCLE 3	`pcircle3 (...)`
CIRCULAR ARC	`pcircular_arc (...)`
CIRCULAR ARC 3	`pcircular_arc3 (...)`
CIRCULAR ARC CLOSE	`pcircular_arc_close (...)`
CIRCULAR ARC CLOSE 3	`pcircular_arc_close3 (...)`
CLEAR TARGET	`pclear_targ (...)`
CLOSE DIRECT INTERPRETATION STRUCTURE	`pclose_di_struct (...)`
CONDITIONAL EXECUTE STRUCTURE	`pcond_exec_struct (...)`
CONDITIONAL INSTANCE STRUCTURE	`pcond_inst_struct (...)`
CONDITIONAL RETURN	`pcond_return (...)`
CONDITIONAL SKIP ELEMENTS	`pcond_skip_elements (...)`
CONDITIONAL SKIP TO LABEL	`pcond_skip_to_label (...)`
COPY ELEMENT FROM STRUCTURE	`pcopy_elem_struct (...)`
COPY ELEMENT RANGE FROM STRUCTURE	`pcopy_elem_range_struct (...)`
COPY ELEMENTS BETWEEN LABELS FROM STRUCTURE	`pcopy_elems_between_labels_struct (...)`
COPY TARGET	`pcopy_targ (...)`
CREATE COMPOSITE MEASURE	`pcreate_composite_measure (...)`
CREATE MIPMAP TEXTURE	`pcreate_mipmap_texture (...)`
CREATE SET MEASURE	`pcreate_set_measure (...)`
CREATE TARGET	`pcreate_targ (...)`
DEFINE CHOICE	`pdefine_choice (...)`
DEFINE COMPOSITE	`pdefine_composite (...)`
DEFINE LINETYPE	`pdefine_linetype (...)`
DEFINE LOCATOR	`pdefine_locator (...)`
DEFINE MARKER TYPE	`pdefine_marker_type (...)`
DEFINE PICK	`pdefine_pick (...)`
DEFINE POSTING GROUP	`pdefine_post_grp (...)`
DEFINE SET	`pdefine_set (...)`
DEFINE STRING	`pdefine_string (...)`
DEFINE STROKE	`pdefine_stroke (...)`
DEFINE VALUATOR	`pdefine_valuator (...)`
DESTROY COMPOSITE MEASURE	`pdestroy_composite_measure (...)`
DESTROY SET MEASURE	`pdestroy_set_measure (...)`

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
DESTROY TARGET	pdestroy_targ (...)
DETACH LOGICAL INPUT DEVICE FROM LIGHT SOURCE	pdetach_lid_from_light_src (...)
DETACH LOGICAL INPUT DEVICE FROM VIEW	pdetach_lid_from_view (...)
DISABLE DIRECT INTERPRETATION PICK	pdisable_di_pick (...)
DISASSOCIATE IMAGE RESOURCE	pdisassoc_image_res (...)
DISASSOCIATE TRAVERSAL RESOURCE	pdisassoc_trav_res (...)
ELLIPSE	pellipse (...)
ELLIPSE 3	pellipse3 (...)
ELLIPTICAL ARC	pelliptical_arc (...)
ELLIPTICAL ARC 3	pelliptical_arc3 (...)
ELLIPTICAL ARC CLOSE	pelliptical_arc_close (...)
ELLIPTICAL ARC CLOSE 3	pelliptical_arc_close3 (...)
ENABLE DIRECT INTERPRETATION PICK	penable_di_pick (...)
END WATCH ON ELEMENT RANGE	pend_watch_on_elem_range (...)
FILL CIRCLE	pfill_circle (...)
FILL CIRCLE 3	pfill_circle3 (...)
FILL ELLIPSE	pfill_ellipse (...)
FILL ELLIPSE 3	pfill_ellipse3 (...)
GET COMPOSITE	pget_composite (...)
GET COMPOSITE 3	pget_composite3 (...)
GET SET	pget_set (...)
GET SET 3	pget_set3 (...)
INITIALIZE COMPOSITE	pinit_composite (...)
INITIALIZE COMPOSITE 3	pinit_composite3 (...)
INITIALIZE DIRECT INTERPRETATION PICK	pinit_di_pick (...)
INITIALIZE DIRECT INTERPRETATION PICK 3	pinit_di_pick3 (...)
INITIALIZE SET	pinit_set (...)
INITIALIZE SET 3	pinit_set3 (...)
INQUIRE ALPHA FACILITIES	pinq_alpha_facs (...)
INQUIRE APPLICATION FILTER	pinq_appl_filter (...)
INQUIRE ASSOCIATED IMAGE RESOURCES	pinq_assoc_image_res (...)
INQUIRE BACK CLIPPING INDICATOR	pinq_back_clip_indicator (...)
INQUIRE BACK PLANE DISTANCE	pinq_back_plane_dist (...)
INQUIRE CHOICE FACILITIES	pinq_choice_facs (...)
INQUIRE COMPOSITE DEVICE STATE	pinq_composite_st (...)
INQUIRE COMPOSITE DEVICE STATE 3	pinq_composite_st3 (...)
INQUIRE COMPOSITE FACILITIES	pinq_composite_facs (...)
INQUIRE COMPOSITE MEASURE DEFINITION	pinq_composite_meas_def (...)
INQUIRE CONDITIONAL TRAVERSAL FACILITIES	pinq_cond_trav_facs (...)
INQUIRE CONFIGURATION SETTING FACILITIES	pinq_config_setting_facs (...)
INQUIRE DEFAULT COMPOSITE DEVICE DATA	pinq_def_composite_data (...)
INQUIRE DEFAULT COMPOSITE DEVICE DATA 3	pinq_def_composite_data3 (...)
INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA	pinq_def_di_pick_data (...)
INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 3	pinq_def_di_pick_data3 (...)
INQUIRE DEFAULT SET DEVICE DATA	pinq_def_set_data (...)
INQUIRE DEFAULT SET DEVICE DATA 3	pinq_def_set_data3 (...)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
INQUIRE DEFAULT TARGET DISPOSITION	pinq_def_targ_disp (...)
INQUIRE DEVICE COORDINATE CLIP REGIONS	pinq_dc_clip_regions (...)
INQUIRE DEVICE COORDINATE CLIP REGIONS 3	pinq_dc_clip_regions3 (...)
INQUIRE DEVICE COORDINATE CLIP REGIONS FACILITIES	pinq_dc_clip_regions_facs(...)
INQUIRE DIRECT INTERPRETATION MODE	pinq_di_mode (...)
INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT	pinq_di_pick_corr_point (...)
INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 3	pinq_di_pick_corr_point3 (...)
INQUIRE DIRECT INTERPRETATION PICK SET STATUS	pinq_di_pick_set_status (...)
INQUIRE DIRECT INTERPRETATION PICK STATE	pinq_di_pick_st (...)
INQUIRE DIRECT INTERPRETATION PICK STATE 3	pinq_di_pick_st3 (...)
INQUIRE DIRECT INTERPRETATION TRAVERSAL FACILITIES	pinq_di_trav_facs (...)
INQUIRE DISPLAY TARGET	pinq_disp_targ (...)
INQUIRE DYNAMICS OF POSTING GROUPS	pinq_dyns_post_grps (...)
INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE	pinq_dyns_ws_attrs_texture (...)
INQUIRE EDGE REPRESENTATION FULL	pinq_edge_rep_full (...)
INQUIRE EXTENDED PATTERN FACILITIES	pinq_ext_pat_facs (...)
INQUIRE EXTENDED PATTERN REPRESENTATION	pinq_ext_pat_rep (...)
INQUIRE FRONT CLIPPING INDICATOR	pinq_front_clip_indicator (...)
INQUIRE FRONT PLANE DISTANCE	pinq_front_plane_dist (...)
INQUIRE HIGHLIGHTING FACILITIES	pinq_highl_facs (...)
INQUIRE HIGHLIGHTING REPRESENTATION	pinq_highl_rep (...)
INQUIRE IMAGE RESOURCE	pinq_image_res (...)
INQUIRE IMAGE RESOURCE FACILITIES	pinq_image_res_facs (...)
INQUIRE LINETYPE DEFINITION	pinq_linetype_def (...)
INQUIRE LINETYPE DEFINITION FACILITIES	pinq_linetype_def_facs (...)
INQUIRE LINETYPE DEFINITION SUPPORT	pinq_linetype_def_support (...)
INQUIRE LIST OF DEFINED APPLICATION FILTERS	pinq_list_def_appl_filters (...)
INQUIRE LIST OF HIGHLIGHTING INDICES	pinq_list_highl_inds (...)
INQUIRE LIST OF POSTING GROUPS	pinq_list_post_grps (...)
INQUIRE LIST OF TEXTURE INDICES	pinq_list_texture_inds (...)
INQUIRE LISTS OF AVAILABLE LOGICAL INPUT DEVICES	pinq_lists_avail_lids (...)
INQUIRE LOCATOR FACILITIES	pinq_loc_facs (...)
INQUIRE LOGICAL INPUT DEVICE ATTACHED TO LIGHT SOURCE	pinq_lid_attached_to_light_src (...)
INQUIRE LOGICAL INPUT DEVICE ATTACHED TO VIEW	pinq_lid_attached_to_view (...)
INQUIRE LOGICAL INPUT DEVICE DEFINITION	pinq_lid_def (...)
INQUIRE MARKER TYPE DEFINITION	pinq_marker_type_def (...)
INQUIRE MARKER TYPE DEFINITION FACILITIES	pinq_marker_type_def_facs (...)
INQUIRE MARKER TYPE DEFINITION SUPPORT	pinq_marker_type_def_support (...)
INQUIRE MIPMAP FACILITIES	pinq_mipmap_facs (...)
INQUIRE NUMBER OF AVAILABLE NON-ATOMIC LOGICAL INPUT DEVICES	pinq_num_avail_na_in (...)
INQUIRE NUMBER OF DEFINED LINETYPES	pinq_num_def_linetypes (...)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
INQUIRE NUMBER OF DEFINED MARKER TYPES	`pinq_num_def_marker_types (...)`
INQUIRE NUMBER OF PREDEFINED APPLICATION FILTERS	`pinq_num_pred_appl_filters (...)`
INQUIRE PICK FACILITIES	`pinq_pick_facs (...)`
INQUIRE PICK MAPPING FACILITIES	`pinq_pick_mapping_facs (...)`
INQUIRE PICK MAPPING STATE	`pinq_pick_mapping_state (...)`
INQUIRE PICTURE STATUS	`pinq_pict_status (...)`
INQUIRE POLYLINE REPRESENTATION FULL	`pinq_line_rep_full (...)`
INQUIRE POSTED DIRECT INTERPRETATION STRUCTURE	`pinq_posted_di_struct (...)`
INQUIRE POSTED STRUCTURES FROM POSTING GROUP	`pinq_posted_structs_from_post_grp (...)`
INQUIRE POSTING GROUP	`pinq_post_grp (...)`
INQUIRE POSTING GROUP FACILITIES	`pinq_post_grp_facs (...)`
INQUIRE PREDEFINED APPLICATION FILTER	`pinq_pred_appl_filter (...)`
INQUIRE PREDEFINED ASSOCIATION OF TARGET WITH TRAVERSAL RESOURCES	`pinq_pred_assoc_targ_trav_res (...)`
INQUIRE PREDEFINED ASSOCIATION OF TRAVERSAL RESOURCE WITH TARGET	`pinq_pred_assoc_trav_res_targ (...)`
INQUIRE PREDEFINED COMPOSITE MEASURE DEFINITION	`pinq_pred_composite_meas_def (...)`
INQUIRE PREDEFINED EDGE REPRESENTATION FULL	`pinq_pred_edge_rep_full (...)`
INQUIRE PREDEFINED EXTENDED PATTERN REPRESENTATION	`pinq_pred_ext_pat_rep (...)`
INQUIRE PREDEFINED HIGHLIGHTING REPRESENTATION	`pinq_pred_highl_rep (...)`
INQUIRE PREDEFINED IMAGE RESOURCE	`pinq_pred_image_res (...)`
INQUIRE PREDEFINED LINETYPE DEFINITION	`pinq_pred_linetype_def (...)`
INQUIRE PREDEFINED MARKER TYPE DEFINITION	`pinq_pred_marker_type_def (...)`
INQUIRE PREDEFINED POLYLINE REPRESENTATION FULL	`pinq_pred_line_rep_full (...)`
INQUIRE PREDEFINED POSTING GROUP	`pinq_pred_post_grp (...)`
INQUIRE PREDEFINED SET MEASURE DEFINITION	`pinq_pred_set_meas_def (...)`
INQUIRE PREDEFINED TEXTURE REPRESENTATION	`pinq_pred_texture_rep (...)`
INQUIRE PROJECTION REFERENCE POINT	`pinq_proj_ref_point (...)`
INQUIRE PROJECTION TYPE	`pinq_proj_type (...)`
INQUIRE PROJECTION VIEWPORT	`pinq_proj_vp (...)`
INQUIRE PROJECTION VIEWPORT 3	`pinq_proj_vp3 (...)`
INQUIRE RENDERING TARGET	`pinq_rend_targ (...)`
INQUIRE SET DEVICE STATE	`pinq_set_st (...)`
INQUIRE SET DEVICE STATE 3	`pinq_set_st3 (...)`
INQUIRE SET FACILITIES	`pinq_set_facs (...)`
INQUIRE SET MEASURE DEFINITION	`pinq_set_meas_def (...)`
INQUIRE SET OF GROUPS TO WHICH POSTED	`pinq_grps_posted (...)`
INQUIRE SET OF WORKSTATIONS TO WHICH DIRECT INTERPRETATION STRUCTURE POSTED	`pinq_wss_di_struct_posted (...)`
INQUIRE STRING FACILITIES	`pinq_string_facs (...)`
INQUIRE STROKE FACILITIES	`pinq_stroke_facs (...)`
INQUIRE TARGET ASSOCIATED WITH TRAVERSAL RESOURCE	`pinq_targ_assoc_trav_res (...)`
INQUIRE TARGET DISPOSITION	`pinq_targ_dispos (...)`
INQUIRE TARGET FACILITIES	`pinq_targ_facs (...)`
INQUIRE TARGET MANIPULATION MODE	`pinq_targ_manip_mode (...)`

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
INQUIRE TARGET STATE	pinq_targ_st (...)
INQUIRE TEXTURE FACILITIES	pinq_texture_facs (...)
INQUIRE TEXTURE MAPPING FACILITIES	pinq_texture_map_facs (...)
INQUIRE TEXTURE REPRESENTATION	pinq_texture_rep (...)
INQUIRE TRANSPARENCY MODE	pinq_transparency_mode (...)
INQUIRE TRANSPARENCY THRESHOLDS	pinq_transparency_thresholds (...)
INQUIRE TRAVERSAL RESOURCE FACILITIES	pinq_trav_res_facs (...)
INQUIRE TRAVERSAL RESOURCES ASSOCIATED WITH TARGET	pinq_trav_res_assoc_targ (...)
INQUIRE VALUATOR FACILITIES	pinq_val_facs (...)
INQUIRE VIEW PLANE DISTANCE	pinq_view_plane_dist (...)
INQUIRE VIEW PLANE NORMAL	pinq_view_plane_norm (...)
INQUIRE VIEW REFERENCE POINT	pinq_view_ref_point (...)
INQUIRE VIEW REFERENCE POINT 3	pinq_view_ref_point3 (...)
INQUIRE VIEW STATUS	pinq_view_status (...)
INQUIRE VIEW UP VECTOR	pinq_view_up_vec (...)
INQUIRE VIEW UP VECTOR 3	pinq_view_up_vec3 (...)
INQUIRE VIEW WINDOW LIMITS	pinq_view_win_limits (...)
INQUIRE WATCH ON ELEMENT RANGE	pinq_watch_elem_range (...)
INQUIRE WATCH RANGE STATE	pinq_watch_range_st (...)
INQUIRE WORKSTATION STATE TABLE LENGTHS HIGHLIGHTING	pinq_ws_st_table_highl (...)
INQUIRE WORKSTATION STATE TABLE LENGTHS TEXTURE	pinq_ws_st_table_texture (...)
INQUIRE X-Y CLIPPING INDICATOR	pinq_xy_clip_indicator (...)
INSTANCE STRUCTURE	pinst_struct (...)
MANIPULATE TRAVERSAL RESOURCE	pmanip_trav_res (...)
MAP DEVICE COORDINATE POINT TO PICK PATHS	pmap_dc_point_to_pick_paths (...)
MAP DEVICE COORDINATES TO WINDOW SYSTEM COORDINATES	pmap_dc_to_wsc (...)
MAP DEVICE COORDINATES TO WORLD COORDINATES	pmap_dc_to_wc (...)
MAP WINDOW SYSTEM COORDINATES TO DEVICE COORDINATES	pmap_wsc_to_dc (...)
MARK MULTI-PASS COMPLETION	pmark_multi_pass_compl (...)
MARK MULTI-PASS START	pmark_multi_pass_start (...)
MARK PASS COMPLETION	pmark_pass_compl (...)
MARK PASS START	pmark_pass_start (...)
MOVE ELEMENT FROM STRUCTURE	pmove_elem_struct (...)
MOVE ELEMENT RANGE FROM STRUCTURE	pmove_elem_range_struct (...)
MOVE ELEMENTS BETWEEN LABELS FROM STRUCTURE	pmove_elems_between_labels_struct (...)
OPEN DIRECT INTERPRETATION STRUCTURE	popen_di_struct (...)
POP STATE	ppop_st (...)
POST DIRECT INTERPRETATION STRUCTURE	ppost_di_struct (...)
POST DIRECT INTERPRETATION STRUCTURE TO POSTING GROUP	ppost_di_struct_to_grp (...)
POST STRUCTURE TO GROUP	ppost_struct_to_grp (...)
PUSH STATE	ppush_st (...)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
REDRAW ALL STRUCTURES FROM POSTING GROUP	predraw_all_structs_from_grp (...)
REDRAW ALL STRUCTURES FROM POSTING GROUP ON TARGET	predraw_all_from_grp_on_targ (...)
REDRAW ALL STRUCTURES ON TARGET	predraw_all_structs_on_targ (...)
RENEW DIRECT INTERPRETATION STATE	prenew_di_state (...)
REQUEST COMPOSITE	preq_composite (...)
REQUEST COMPOSITE 3	preq_composite3 (...)
REQUEST SET	preq_set (...)
REQUEST SET 3	preq_set3 (...)
RESET ALL TRAVERSAL RESOURCES	preset_all_trav_res (...)
RETRIEVE NUMBER OF PASSES REQUIRED	pret_num_passes_req (...)
RETRIEVE WINDOW SYSTEM COLOUR	pret_window_system_colr (...)
SAMPLE COMPOSITE	psample_composite (...)
SAMPLE COMPOSITE 3	psample_composite3 (...)
SAMPLE SET	psample_set (...)
SAMPLE SET 3	psample_set3 (...)
SET ACTIVE TEXTURES	pset_active_textures (...)
SET ALPHA DATA SELECTION INDEX	pset_alpha_data_sel_ind (...)
SET ALPHA SOURCE SELECTOR	pset_alpha_src_sel (...)
SET APPLICATION FILTER	pset_appl_filter (...)
SET APPLICATION INTEGER	pset_appl_int (...)
SET APPLICATION REAL	pset_appl_real (...)
SET BACK ACTIVE TEXTURES	pset_back_active_textures (...)
SET BACK CLIPPING INDICATOR	pset_back_clip_indicator (...)
SET BACK PLANE DISTANCE	pset_back_plane_dist (...)
SET BACK TRANSPARENCY	pset_back_transparency (...)
SET COMPOSITE MODE	pset_composite_mode (...)
SET COMPOSITE PICK FILTER	pset_composite_pick_filter (...)
SET CONDITION FLAGS	pset_cond_flags (...)
SET CONDITION FLAGS FROM TESTS	pset_cond_flags_from_tests (...)
SET DEPTH CUE REPRESENTATION MASK	pset_depth_cue_rep_mask (...)
SET DEVICE COORDINATE CLIP REGIONS	pset_dc_clip_regions (...)
SET DEVICE COORDINATE CLIP REGIONS 3	pset_dc_clip_regions3 (...)
SET DIRECT INTERPRETATION MODE	pset_di_mode (...)
SET DIRECT INTERPRETATION PICK CORRELATION POINT	pset_di_pick_corr_point (...)
SET DIRECT INTERPRETATION PICK CORRELATION POINT 3	pset_di_pick_corr_point3 (...)
SET DIRECT INTERPRETATION PICK FILTER	pset_di_pick_filter (...)
SET DISPLAY TARGET	pset_disp_targ (...)
SET EDGECAP	pset_edgecap (...)
SET EDGEJOIN	pset_edgejoin (...)
SET EDGEMITRE LIMIT	pset_edgemitre_limit (...)
SET EDGE REPRESENTATION FULL	pset_edge_rep_full (...)
SET EDGE REPRESENTATION MASK	pset_edge_rep_mask (...)
SET EDGETYPE ADAPTABILITY	pset_edgetype_adapt (...)
SET EDGETYPE CONTINUITY	pset_edgetype_cont (...)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
SET EDGETYPE OFFSET	pset_edgetype_offset (...)
SET EXTENDED PATTERN REPRESENTATION	pset_ext_pat_rep (...)
SET FRONT CLIPPING INDICATOR	pset_front_clip_indicator (...)
SET FRONT PLANE DISTANCE	pset_front_plane_dist (...)
SET HIGHLIGHTING INDEX	pset_highl_ind (...)
SET HIGHLIGHTING METHOD	pset_highl_method (...)
SET HIGHLIGHTING REPRESENTATION	pset_highl_rep (...)
SET INTERIOR REPRESENTATION MASK	pset_int_rep_mask (...)
SET LINECAP	pset_linecap (...)
SET LINEJOIN	pset_linejoin (...)
SET LINEMITRE LIMIT	pset_linemitre_limit (...)
SET LINETYPE ADAPTABILITY	pset_linetype_adapt (...)
SET LINETYPE CONTINUITY	pset_linetype_cont (...)
SET LINETYPE OFFSET	pset_linetype_offset (...)
SET PATTERN REPRESENTATION MASK	pset_pat_rep_mask (...)
SET PICK MAPPING DATA	pset_pick_mapping_data (...)
SET POLYLINE REPRESENTATION FULL	pset_line_rep_full (...)
SET POLYLINE REPRESENTATION MASK	pset_line_rep_mask (...)
SET POLYMARKER REPRESENTATION MASK	pset_marker_rep_mask (...)
SET POSTING GROUP BACKGROUND METHOD	pset_post_grp_backg_method (...)
SET POSTING GROUP BACKGROUND STYLE	pset_post_grp_backg_style (...)
SET POSTING GROUP BORDER INDEX	pset_post_grp_border_ind (...)
SET POSTING GROUP BORDER INDICATOR	pset_post_grp_border_indicator (...)
SET POSTING GROUP PRIORITY	pset_post_grp_priority (...)
SET POSTING GROUP STATUS	pset_post_grp_status (...)
SET PROJECTION REFERENCE POINT	pset_proj_ref_point (...)
SET PROJECTION TYPE	pset_proj_type (...)
SET PROJECTION VIEWPORT	pset_proj_vp (...)
SET PROJECTION VIEWPORT 3	pset_proj_vp3 (...)
SET REFLECTANCE REPRESENTATION MASK	pset_refl_rep_mask (...)
SET RENDERING TARGET	pset_rend_targ (...)
SET SET MODE	pset_set_mode (...)
SET SET PICK FILTER	pset_set_pick_filter (...)
SET STATE OF VISUAL REPRESENTATION	pset_st_visual_rep (...)
SET TARGET DISPOSITION	pset_targ_dispos (...)
SET TARGET MANIPULATION MODE	pset_targ_manip_mode (...)
SET TARGET STATE OF VISUAL REPRESENTATION	pset_targ_st_visual_rep (...)
SET TEXT REPRESENTATION MASK	pset_text_rep_mask (...)
SET TEXTURE BINDING	pset_texture_binding (...)
SET TEXTURE COMPOSITION	pset_texture_composition (...)
SET TEXTURE PARAMETRIZATION	pset_texture_param (...)
SET TEXTURE PERSPECTIVE CORRECTION	pset_texture_perspect_corr (...)
SET TEXTURE REPRESENTATION	pset_texture_rep (...)
SET TEXTURE REPRESENTATION MASK	pset_texture_rep_mask (...)
SET TEXTURE RESOURCE OPTIMIZATION HEURISTICS	pset_texture_res_opt_heur (...)
SET TEXTURE SAMPLING	pset_texture_sampling (...)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name (Continued)

PHIGS and PHIGS PLUS Name	C Name
SET TEXTURE SAMPLING FREQUENCY	`pset_texture_sampling_freq (...)`
SET TRANSPARENCY	`pset_transparency (...)`
SET TRANSPARENCY MODE	`pset_transparency_mode (...)`
SET TRANSPARENCY THRESHOLDS	`pset_transparency_thresholds (...)`
SET VIEW PLANE DISTANCE	`pset_view_plane_dist (...)`
SET VIEW PLANE NORMAL	`pset_view_plane_norm (...)`
SET VIEW REFERENCE POINT	`pset_view_ref_point (...)`
SET VIEW REFERENCE POINT 3	`pset_view_ref_point3 (...)`
SET VIEW UP VECTOR	`pset_view_up_vec (...)`
SET VIEW UP VECTOR 3	`pset_view_up_vec3 (...)`
SET VIEW WINDOW LIMITS	`pset_view_win_limits (...)`
SET WATCH ON ELEMENT RANGE	`pset_watch_on_elem_range (...)`
SET X-Y CLIPPING INDICATOR	`pset_xy_clip_indicator (...)`
TRANSFER DIRECT INTERPRETATION PICK SET	`ptrans_di_pick_set (...)`
UNDEFINE CHOICE	`pundefine_choice (...)`
UNDEFINE COMPOSITE	`pundefine_composite (...)`
UNDEFINE LOCATOR	`pundefine_locator (...)`
UNDEFINE PICK	`pundefine_pick (...)`
UNDEFINE POSTING GROUP	`pundefine_post_grp (...)`
UNDEFINE SET	`pundefine_set (...)`
UNDEFINE STRING	`pundefine_string (...)`
UNDEFINE STROKE	`pundefine_stroke (...)`
UNDEFINE VALUATOR	`pundefine_valuator (...)`
UNPOST ALL STRUCTURES FROM GROUP	`punpost_all_structs_from_grp (...)`
UNPOST DIRECT INTERPRETATION STRUCTURE	`punpost_di_struct (...)`
UNPOST STRUCTURES FROM GROUPS	`punpost_structs_from_grps (...)`
UPDATE TARGET	`pupd_targ (...)`
UPDATE VIEW REPRESENTATION	`pupd_view_rep (...)`
WORKSTATION TYPE CREATE	`pws_type_create (...)`
WORKSTATION TYPE DESTROY	`pws_type_destroy (...)`
WORKSTATION TYPE GET	`pws_type_get (...)`
WORKSTATION TYPE SET	`pws_type_set (...)`

"

Page 27

5 C PHIGS type definitions

*Change the clause heading to "***C *Basic PHIGS profile* type definitions***".*

5.1 Mapping of PHIGS data types

*Change the subclause heading to "***Mapping of *Basic PHIGS profile* data types***".*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The following text replaces the text of the first paragraph:

"The *Basic PHIGS profile* specifies a set of abstract data types. Table 4 gives the mapping from those data types defined in the *Basic PHIGS profile*."

Change the table title to "**Table - 4 Mapping of *Basic PHIGS profile* data types to C**".

Change the left table heading to "***Basic PHIGS profile* data type**".

Page 27

5.2 Environmental type definitions

Append the phrase "of all profiles" *to the second sentence of the first paragraph.*

Page 28

5.3 Implementation dependent type definitions

Change the subclause heading to "**Implementation dependent *Basic PHIGS profile* type definitions**".

The following text replaces the phrases of the "xxx" description concerned with the escape data records.

', "escape_in" for ESCAPE input data records, "escape_out" for ESCAPE output data records, "targ_op" for target operation types, "trav_res_op" for traversal resource operations, "in_type" for new classes of input measures'

Pages 29 and 30

The following type definitions replace the Pchoice_data *and* Pchoice_data3 *type definitions:*

"

Pchoice_data **CHOICE DATA RECORD**

```
typedef struct {
   union Pchoice_pets {
      struct Pchoice_pet_other {
         Pint      unused;
      } pet_other; /* When no echo-specific data is required */
      struct Pchoice_pet_r1{
         . . .     /* impl. dependent */
      } pet_r1;  /* For PET 1 */
      struct Pchoice_pet_r2 {
         Pint         num_prompts;     /* number of prompts*/
         Ppr_switch   *prompts;        /* array of prompts*/
         . . .                         /* impl. defined*/
      } pet_r2;       /* For PET 2 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Pchoice_pet_r3 {
         Pint      num_strings;        /* number of choice strings*/
         char      **strings;          /* array of choice strings*/
         . . .                         /* impl. defined*/
      } pet_r3;  /* For PET 3 */
      struct Pchoice_pet_r4 {
         Pint      num_strings;        /* number of alternatives*/
         char      **strings;          /* array of strings*/
         . . .                         /* impl. defined*/
      } pet_r4;  /* For PET 4 */
      struct Pchoice_pet_r5 {
         Pint      struct_id;          /* structure identifier*/
         Pint      num_pick_ids;       /* number of alternatives*/
         Pint      *pick_ids;          /* array of pick identifiers*/
         . . .                         /* impl. defined*/
      } pet_r5;  /* For PET 5 */
      . . .        /* data for impl. defined pets */
   } pets;
   union Pchoice_measure_data {
      struct Pchoice_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Pchoice_trigger_data {
      struct Pchoice_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined trigger processes */
   } trigger_data;
   union Pchoice_ack_data {
      struct Pchoice_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pchoice_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pchoice_data3 **CHOICE DATA RECORD 3**

```
typedef struct {
   union Pchoice3_pets {
      struct Pchoice3_pet_other {
         Pint      unused;
      } pet_other; /* When no echo-specific data is required */
      struct Pchoice3_pet_r1{
         . . .                              /* impl. dependent*/
      } pet_r1;  /* For PET 1 */
      struct Pchoice3_pet_r2 {
         Pint      num_prompts;             /* number of prompts*/
         Ppr_switch*prompts;                /* array of prompts*/
         . . .                              /* impl. defined*/
      } pet_r2;       /* For PET 2 */
      struct Pchoice3_pet_r3 {
         Pint      num_strings;             /* number of choice strings*/
         char      **strings;               /* array of choice strings*/
         . . .                              /* impl. defined*/
      } pet_r3;  /* For PET 3 */
      struct Pchoice3_pet_r4 {
         Pint      num_strings;             /* number of alternatives*/
         char      **strings;               /* array of strings*/
         . . .                              /* impl. defined*/
      } pet_r4;  /* For PET 4 */
      struct Pchoice3_pet_r5 {
         Pint      struct_id;               /* structure identifier*/
         Pint      num_pick_ids;            /* number of alternatives*/
         Pint      *pick_ids;               /* array of pick identifiers*/
         . . .                              /* impl. defined*/
      } pet_r5;  /* For PET 5 */
      . . .        /* data for impl. defined pets */
   } pets;
   union Pchoice3_measure_data {
      struct Pchoice_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    union Pchoice3_trigger_data {
       struct Pchoice3_trig_other {
          Pint      unused;
       } trig_other; /* When no trigger-specific data is required */
       . . .        /* data for impl. defined trigger processes */
    } trigger_data;
    union Pchoice3_ack_data {
       struct Pchoice3_ack_other {
          Pint      unused;
       } ack_other;  /* When no ack.-specific data is required */
       . . .        /* data for impl. defined ack. processes */
    } acknowledgement_data;
 } Pchoice_data3;"
```

Pages 33 and 34

The following type definition replaces the definitions of Ploc_data and Ploc_data3 respectively:

"

Ploc_data **LOCATOR DATA RECORD**

```
 typedef struct {
    union Ploc_pets {
       struct Ploc_pet_other {
          Pint      unused;
       } pet_other;    /* When no echo-specific data is required */
       struct Ploc_pet_r1 {
          . . .                          /* impl. dependent*/
       } pet_r1;       /* Supports standard PET 1 */
       struct Ploc_pet_r2 {
          . . .                          /* impl. dependent*/
       } pet_r2;       /* Supports standard PET 2 */
       struct Ploc_pet_r3 {
          . . .                          /* impl. dependent*/
       } pet_r3;       /* Supports standard PET 3 */
       struct Ploc_pet_r4 {
          Pline_attrs line_attrs;        /* polyline attributes*/
          . . .                          /* impl. dependent*/
       } pet_r4;       /* Supports standard PET 4 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Ploc_pet_r5 {
         Pline_fill_ctrl_flag  line_fill_ctrl_flag;   /* ctrl. flag*/
         union Ploc_attrs {
            Pline_attrs   line_attrs; /* polyline attributes*/
            Pint_attrs    int_attrs;  /* interior attributes*/
            struct Ploc_fill_set {
               Pint_attrs  int_attrs;  /* interior attributes*/
               Pedge_attrs edge_attrs; /* edge attributes*/
            } fill_set;
         } attrs;
         . . .          /* data for impl. defined PET's */
      } pet_r5;         /* Supports standard PET 5 */
      struct Ploc_pet_r6 {
         . . .                          /* impl. dependent*/
      } pet_r6;         /* Supports standard PET 6 */
      . . .             /* data for impl. defined PET's */
   } pets;
   union Ploc_measure_data {
      struct Ploc_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .         /* data for impl. defined measure processes */
   } measure_data;
   union Ploc_trigger_data{
      struct Ploc_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .         /* data for impl. defined trigger processes */
   } trigger_data;
   union Ploc_ack_data {
      struct Ploc_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      . . .         /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Ploc_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Ploc_data3 **LOCATOR DATA RECORD 3**

```
typedef struct {
   union Ploc3_pets {
      struct Ploc3_pet_other {
         Pint      unused;
      } pet_other;    /* When no echo-specific data is required */
      struct Ploc3_pet_r1 {
         . . .                          /* impl. dependent*/
      } pet_r1;       /* Supports standard PET 1 */
      struct Ploc3_pet_r2 {
         . . .                          /* impl. dependent*/
      } pet_r2;       /* Supports standard PET 2 */
      struct Ploc3_pet_r3 {
         . . .                          /* impl. dependent*/
      } pet_r3;       /* Supports standard PET 3 */
      struct Ploc3_pet_r4 {
         Pline_attrs line_attrs;        /* polyline attributes*/
           . . .                        /* impl. dependent*/
      } pet_r4;       /* Supports standard PET 4 */
      struct Ploc3_pet_r5 {
         Pline_fill_ctrl_flag line_fill_ctrl_flag;  /* ctrl. flag*/
         union Ploc3_attrs {
            Pline_attrs   line_attrs; /* polyline attributes*/
            Pint_attrs    int_attrs;  /* interior attributes*/
            struct Ploc3_fill_set {
               Pint_attrs  int_attrs;   /* interior attributes*/
               Pedge_attrs edge_attrs;  /* edge attributes*/
            } fill_set;
         } attrs;
         . . .          /* data for impl. defined PET's */
      } pet_r5;       /* Supports standard PET 5 */
      struct Ploc3_pet_r6 {
         . . .                          /* impl. dependent*/
      } pet_r6;       /* Supports standard PET 6 */
      . . .         /* data for impl. defined PET's */
   } pets;
   union Ploc3_measure_data {
      struct Ploc3_meas_other {
         Pint      unused;
      } meas_other;  /* When no measure-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      . . .          /* data for impl. defined measure processes */
   } measure_data;
   union Ploc3_trigger_data {
      struct Ploc3_trig_other {
         Pint        unused;
      } trig_other;  /* When no trigger-specific data is required */
      . . .          /* data for impl. defined trigger processes */
   } trigger_data;
   union Ploc3_ack_data {
      struct Ploc3_ack_other {
         Pint        unused;
      } ack_other;   /* When no ack.-specific data is required */
      . . .          /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Ploc_data3;"
```

Pages 35 through 38

The following type definitions replace the corresponding definitions respectively:

"

Ppick_data **PICK DATA RECORD**

```
typedef struct {
   union Ppick_pets {
      struct Ppick_pet_other {
         Pint        unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Ppick_pet_r1 {
         . . .                          /* impl. dependent*/
      } pet_r1;  /* For PET 1 */
      . . .          /* data for impl. defined pets */
      struct Ppick_pet_r2 {
         ...                            /* implementation dependent*/
      } pet_r2;  /* For PET 2 */
      struct Ppick_pet_r3 {
         ...                            /* implementation dependent*/
      } pet_r3;  /* For PET 3 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Ppick_pet_r4 {
         Pint      highl_ind;          /* highlighting index*/
         . . .                         /* impl. dependent*/
      } pet_r4;  /* For PET 4 */
      struct Ppick_pet_r5 {
         Pint      highl_ind;          /* highlighting index*/
         . . .                         /* impl. dependent*/
      } pet_r5;  /* For PET 5 */
      struct Ppick_pet_r6 {
         Pint      highl_ind;          /* highlighting index*/
         . . .                         /* impl. dependent*/
      } pet_r6;  /* For PET 6 */
      ...                              /* implementation defined PET's*/
   } pets;
   union Ppick_measure_data {
      struct Ppick_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Ppick_trigger_data {
      struct Ppick_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined pick processes */
   } trigger_data;
   union Ppick_ack_data {
      struct Ppick_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Ppick_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Ppick_data3 **PICK DATA RECORD 3**

```
typedef struct {
   union Ppick3_pets {
      struct Ppick3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Ppick3_pet_r1 {
         . . .                              /* impl. dependent*/
      } pet_r1;  /* For PET 1 */
      struct Ppick3_pet_r2 {
         ...                                /* implementation dependent*/
      } pet_r2;  /* For PET 2 */
      struct Ppick3_pet_r3 {
         ...                                /* implementation dependent*/
      } pet_r3;  /* For PET 3 */
      struct Ppick3_pet_r4 {
         Pint      highl_ind;               /* highlighting index*/
         . . .                              /* impl. dependent*/
      } pet_r4;  /* For PET 4 */
      struct Ppick3_pet_r5 {
         Pint      highl_ind;               /* highlighting index*/
         . . .                              /* impl. dependent*/
      } pet_r5;  /* For PET 5 */
      struct Ppick3_pet_r6 {
         Pint      highl_ind;               /* highlighting index*/
         . . .                              /* impl. dependent*/
      } pet_r6;
      . . .        /* data for impl. defined pets */
   } pets;  /* For PET 6 */
   union Ppick3_measure_data {
      struct Ppick3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Ppick3_trigger_data {
      struct Ppick3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    . . .        /* data for impl. defined pick processes */
  } trigger_data;
  union Ppick3_ack_data {
    struct Ppick3_ack_other {
      Pint      unused;
    } ack_other; /* When no ack.-specific data is required */
    . . .        /* data for impl. defined ack. processes */
  } acknowledgement_data;
} Ppick_data3;
```

Pstring_data **STRING DATA RECORD**

```
typedef struct {
  Pint              in_buf_size;    /* input buffer size*/
  Pint              init_pos;       /* initial editing position*/
  union Pstring_pets {
    struct Pstring_pet_other {
      Pint      unused;
    } pet_other;  /* When no echo-specific data is required */
    struct Pstring_pet_r1{
      . . .                         /* impl. dependent*/
    } pet_r1;  /* For PET 1 */
    . . .        /* data for impl. defined pets */
  } pets;
  union Pstring_measure_data {
    struct Pstring_meas_other {
      Pint      unused;
    } meas_other; /* When no measure-specific data is required */
    . . .        /* data for impl. defined measure processes */
  } measure_data;
  union Pstring_trigger_data {
    struct Pstring_trig_other {
      Pint      unused;
    } trig_other; /* When no trigger-specific data is required */
    . . .        /* data for impl. defined trigger processes */
  } trigger_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    union Pstring_ack_data {
       struct Pstring_ack_other {
          Pint      unused;
       } ack_other;  /* When no ack.-specific data is required */
       . . .        /* data for impl. defined ack. processes */
    } acknowledgement_data;

} Pstring_data;
```

Pstring_data3 **STRING DATA RECORD 3**

```
typedef struct {
   Pint                in_buf_size;    /* input buffer size*/
   Pint                init_pos;       /* initial editing position*/
   union Pstring3_pets {
      struct Pstring_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstring3_pet_r1{
         . . .                         /* impl. dependent*/
      } pet_r1;  /* For PET 1 */
      . . .        /* data for impl. defined pets */
   } pets;
   union Pstring3_measure_data {
      struct Pstring3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Pstring3_trigger_data {
      struct Pstring3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined trigger processes */
   } trigger_data;
   union Pstring3_ack_data {
      struct Pstring3_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pstring_data3;
```

Pstroke_data **STROKE DATA RECORD**

```
typedef struct {
   Pint              buffer_size;    /* input buffer size */
   Pint              init_pos;       /* initial editing position */
   Pfloat            x_interval;     /* x trigger interval */
   Pfloat            y_interval;     /* y trigger interval */
   Pfloat            time_interval; /* time trigger interval */
   union {
      struct Pstroke_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstroke_pet_r1 {
         . . .                          /* impl. dependent*/
      } pet_r1;  /* For std PET 1 */
      struct Pstroke_pet_r2 {
         . . .                          /* impl. dependent*/
      } pet_r2;  /* For std PET 2 */
      struct Pstroke_pet_r3 {
         Pmarker_attrs  marker_attrs; /* marker attributes*/
         . . .                          /* impl. dependent*/
      } pet_r3;  /* For std PET 3 */
      struct Pstroke_pet_r4 {
         Pline_attrsline_attrs;         /* line attributes*/
         . . .                          /* impl. dependent*/
      } pet_r4;  /* For std PET 4 */
      . . .        /* data for impl. defined PET's */
         } pets;
   union Pstroke_measure_data {
      struct Pstroke_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pstroke_trigger_data {
      struct Pstroke_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined trigger processes */
   } trigger_data;

   union Pstroke_ack_data {
      struct Pstroke_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pstroke_data;
```

Pstroke_data3 **STROKE DATA RECORD 3**

```
typedef struct {
   Pint              in_buf_size;    /* input buffer size*/
   Pint              init_pos;       /* initial editing position*/
   Pfloat            x_interval;     /* x trigger interval*/
   Pfloat            y_interval;     /* y trigger interval*/
   Pfloat            z_interval;     /* z trigger interval*/
   Pfloat            time_interval; /* time trigger interval*/
   union Pstroke3_pets {
      struct Pstroke3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstroke3_pet_r1 {
         . . .                          /* impl. dependent*/
      } pet_r1;  /* For std PET 1 */
      struct Pstroke3_pet_r2 {
         . . .     /* impl. dependent */
      } pet_r2;  /* For std PET 2 */
      struct Pstroke3_pet_r3 {
         Pmarker_attrs  marker_attrs; /* marker attributes*/
         . . .                        /* impl. dependent*/
      } pet_r3;  /* For std PET 3 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Pstroke3_pet_r4 {
         Pline_attrs   line_attrs;     /* line attributes*/
         . . .                         /* impl. dependent*/
      } pet_r4;  /* For std PET 4 */
      . . .        /* data for impl. defined PET's */
   } pets;
   union Pstroke3_measure_data {
      struct Pstroke3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Pstroke3_trigger_data {
      struct Pstroke3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined trigger processes */
   } trigger_data;
   union Pstroke3_ack_data {
      struct Pstroke3_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pstroke_data3;
```

Pval_data VALUATOR DATA RECORD

```
typedef struct {
   Pfloat              low;          /* Low value of valuator range*/
   Pfloat              high;         /* High value of valuator range*/
   union Pval_pets {
      struct Pval_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      . . .        /* data for impl. defined pets */
   } pets;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pval_measure_data {
      struct Pval_measure_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Pval_trigger_data {
      struct Pval_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined trigger processes */
   } trigger_data;
   union Pval_ack_data {
      struct Pval_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pval_data;
```

Pval_data3 **VALUATOR DATA RECORD 3**

```
typedef struct {
   Pfloat              low;          /* Low value of valuator range*/
   Pfloat              high;         /* High value of valuator range*/
   union Pval3_pets {
      struct Pval3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      . . .        /* data for impl. defined pets */
   } pets;
   union Pval3_measure_data {
      struct Pval3_measure_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    union Pval3_trigger_data {
       struct Pval3_trig_other {
          Pint      unused;
       } trig_other; /* When no trigger-specific data is required */
       . . .        /* data for impl. defined trigger processes */
    } trigger_data;
    union Pval3_ack_data {
       struct Pval3_ack_other {
          Pint      unused;
       } ack_other; /* When no ack.-specific data is required */
       . . .        /* data for impl. defined ack. processes */
    } acknowledgement_data;
} Pval_data3;"
```

Page 38

5.4 Implementation independent type definitions

*Change the subclause heading to "***Implementation independent *Basic PHIGS profile* type definitions***".*

Page 65

The following text replaces item PSTRUCT_ST_STOP *in the definition of Pstruct_st:*

```
"PSTRUCT_ST_STOP,
 PSTRUCT_ST_DISO"
```

Page 73

6 C PHIGS macro definitions

*Change the clause heading to "***C *Basic PHIGS profile* macro definitions***".*

Page 83

6.3 Linetypes

6.3.1 Linetypes

The following entry is appended:

```
"
#define PLINE_DASH_DOT_DOT                    (5)"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Page 76

6.4 Error codes

The following definition replaces error 5:

"
```
#define PE_NOT_STOP            (5)    /* Ignoring function, function requires
                                         state (PHOP,*,STOP | DISO,*) */"
```

Page 85

7 C PHIGS functions

*Change the clause heading to "***C *Basic PHIGS profile* functions***".*

Page 17 (ISO/IEC 9593-4:1992/Amd. 1:1994)

8 C PHIGS PLUS type definitions

*Change the subclause heading to "***C *PHIGS PLUS profile* type definitions***".*

8.1 Mapping of PHIGS PLUS data types

*Change the clause heading to "***Mapping of *PHIGS PLUS profile* data types***".*

The following text replaces the text of the first paragraph:

"The *PHIGS PLUS profile* specifies a set of abstract data types beyond the types defined in the *Basic PHIGS profile*. Table 4 gives the mapping from those additional data types defined in the *PHIGS PLUS profile*."

*Change the table title to "***Table - 4 Mapping of *PHIGS PLUS profile* data types to C***".***

*Change the left table heading to "****PHIGS PLUS profile* data type***".*

Page 18 (ISO/IEC 9593-4:1992/Amd. 1:1994)

8.2 Modifications to PHIGS data types

*Change the subclause heading to "***Modifications to *Basic PHIGS profile* data types***".*

Page 24 (ISO/IEC 9593-4:1992/Amd. 1:1994)

8.3 Implementation dependent PHGIS PLUS type definitions

*Change the subclause heading to "***Implementation dependent *PHIGS PLUS profile* type definitions***".*

Page 34 (ISO/IEC 9593-4:1992/Amd. 1:1994)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8.4 Implementation independent PHIGS PLUS type definitions

*Change the subclause heading to "***Implementation independent *PHIGS PLUS profile* type definitions***".*

Page 56 (ISO/IEC 9593-4:1992/Amd. 1:1994)

9 C PHIGS PLUS macro definitions

*Change the clause heading to "***C *PHIGS PLUS profile* macro definitions***".*

Page 63 (ISO/IEC 9593-4:1992/Amd. 1:1994)

10 C PHIGS PLUS functions

*Change the clause heading to "***C *PHIGS PLUS profile* functions***".*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Page 95 (ISO/IEC 9593-4:1992/Amd. 2:1994)

The following new clauses are inserted following clause 10.

"

11 C *Full PHIGS profile* type definitions

11.1 Mapping of *Full PHIGS profile* data types

The *Full PHIGS profile* specifies a set of abstract data types beyond the types defined in the *PHIGS PLUS profile*. Table 4 gives the mapping from those additional data types defined in the *Full PHIGS profile*.

Table 4 - Mapping of *Full PHIGS profile* data types to C

PHIGS data type		C binding data type
@t	location of t	*t
AP	acknowledgement process	Pack_process
BIT	bit	unsigned int:1
CF	configuration setting list	Pconfig_settings
CFT	condition flag test	Pcond_test
EP	echo process	Pecho_process
M	measure	Pmeasure
MD	marker descriptor	Pmarker_desc
MP	measure process	Pmeas_process
MS	measure set	Pmeasure_set
TA	target address	Ptarg_addr
TO	target operation	Ptarg_op
TP	trigger process	Ptrig_process
TR	traversal resource	Ptrav_res

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

11.2 Modifications to *Basic PHIGS profile* type definitions

`Paspect` **ASPECT**

```
typedef enum {
   /* start of PHIGS enumerations */
   ...,
   /* end of PHIGS enumerations */
   /* start of PHIGS PLUS enumerations */
   ...,
   /* end of PHIGS PLUS enumerations */
   /* start of PHIGS FULL enumerations */
   PASPECT_HIGHL_METHOD
   /* end of PHIGS FULL enumerations */
} Paspect;
```

`Pcolr_rep` **COLOUR REPRESENTATION**

```
typedef union {
   /* start of Basic PHIGS union members */
   Prgb      rgb;     /* RGB colour specification                    */
   Pcieluv   cieluv;  /* CIE L*u*v* colour specification             */
   Phls      hls;     /* HLS colour specification                    */
   Phsv      hsv;     /* HSV colour specification                    */
   /* end of Basic PHIGS union members */
   /* start of Full PHIGS union members */
   Prgba     rgba;    /* RGB specification with alpha                */
   Pcieluva  cieluva; /* CIE L*u*v* colour specfication with alpha   */
   Phlsa     hlsa;    /* HLS colour specificaiton with alpha         */
   Phsva     hsva;    /* HSV colour specification with alpha         */
   /* end of Full PHIGS union members */
   ...                /* implementation defined */
} Pcolr_rep;
```

`Pelem_data` **ELEMENT DATA**

```
typedef union {
  /* start of Basic PHIGS element data */
  ...
  /* end of Basic PHIGS element data */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* start of PHIGS PLUS element data */
...
/* end of PHIGS PLUS element data */
/* start of Full PHIGS element data */
struct Pappl_int {
  Pint   integer_id;     /* application integer identifier          */
  Pint   value;          /* value for application integer           */
} appl_int;
struct Pappl_real {
  Pint      real_id;        /* application real identifier          */
  Pfloat    value;          /* value for application real           */
} appl_real;
struct Pcircle3 {
    Ppoint3      center_point;       /* center point                 */
    Pfloat       radius;             /* radius                       */
    Pvec3        ref_vecs[2];        /* reference vectors            */
} circle3;
struct Pcircle {
  Ppoint       center_point;         /* center point                 */
  Pfloat       radius;               /* radius                       */
} circle;
struct Pcircular_arc3 {
  Ppoint3             center_point;  /* center point                 */
  Pfloat              radius;        /* radius                       */
  Pvec3               ref_vecs[2];   /* reference vectors            */
  Pfloat              start;         /* start angle                  */
  Pfloat              end;           /* end angle                    */
} circular_arc3;
struct Pcircular_arc {
  Ppoint              center_point;  /* center point                 */
  Pfloat              radius;        /* radius                       */
  Pfloat              start;         /* start angle                  */
  Pfloat              end;           /* end angle                    */
} circular_arc;
struct Pellipse3 {
  Ppoint3         center_point;      /* center point                 */
  Pvec3           major_ref_vec;     /* major axis reference vector */
  Pvec3           minor_ref_vec;     /* minor axis reference vector */
} ellipse3;
struct Pellipse {
  Ppoint          center_point;      /* center point                 */
  Pvec            major_ref_vec;     /* major axis reference vector */
  Pvec            minor_ref_vec;     /* minor axis reference vector */
} ellipse;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
struct Pelliptical_arc3 {
  Ppoint3         center_point;      /* center point                */
  Pvec3           major_ref_vec;     /* major axis reference vector */
  Pvec3           minor_ref_vec;     /* minor axis reference vector */
  Pfloat          start;             /* start angle                 */
  Pfloat          end;               /* end angle                   */
} elliptical_arc3;

struct Pelliptical_arc {
  Ppoint          center_point;      /* center point                */
  Pvec            major_ref_vec;     /* major axis reference vector */
  Pvec            minor_ref_vec;     /* minor axis reference vector */
  Pfloat          start;             /* start angle                 */
  Pfloat          end;               /* end angle                   */
} elliptical_arc;

struct Pfill_circle3 {
  Ppoint3      center_point;         /* center point                */
  Pfloat       radius;               /* radius                      */
  Pvec3        ref_vecs[2];          /* reference vectors           */
} fill_circle3;

struct Pfill_circle {
  Ppoint       center_point;         /* center point                */
  Pfloat       radius;               /* radius                      */
} fill_circle;

struct Pcircular_arc_close3 {
  Ppoint3             center_point;  /* center point                */
  Pfloat              radius;        /* radius                      */
  Pvec3               ref_vecs[2];   /* reference vectors           */
  Pfloat              start;         /* start angle                 */
  Pfloat              end;           /* end angle                   */
  Pclosure            type;          /* closure type                */
} circular_arc_close3;

struct Pcircular_arc_close {
  Ppoint              center_point;  /* center point                */
  Pfloat              radius;        /* radius                      */
  Pfloat              start;         /* start angle                 */
  Pfloat              end;           /* end angle                   */
  Pclosure            type;          /* closure type                */
} circular_arc_close;

struct Pfill_ellipse3 {
  Ppoint3         center_point;      /* center point                */
  Pvec3           major_ref_vec;     /* major axis reference vector */
  Pvec3           minor_ref_vec;     /* minor axis reference vector */
} fill_ellipse3;

struct Pfill_ellipse {
  Ppoint          center_point;      /* center point                */
  Pvec            major_ref_vec;     /* major axis reference vector */
  Pvec            minor_ref_vec;     /* minor axis reference vector */
} fill_ellipse;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
struct Pelliptical_arc_close3 {
  Ppoint3         center_point;       /* center point                */
  Pvec3           major_ref_vec;      /* major axis reference vector */
  Pvec3           minor_ref_vec;      /* minor axis reference vector */
  Pfloat          start;              /* start angle                 */
  Pfloat          end;                /* end angle                   */
  Pclosure        type;               /* closure type                */
} elliptical_arc_close3;
struct Pelliptical_arc_close {
  Ppoint          center_point;       /* center point                */
  Pvec            major_ref_vec;      /* major axis reference vector */
  Pvec            minor_ref_vec;      /* minor axis reference vector */
  Pfloat          start;              /* start angle                 */
  Pfloat          end;                /* end angle                   */
  Pclosure        type;               /* closure type                */
} elliptical_arc_close;
Plinetype_adapt  linetype_adapt;     /* linetype adaptability       */
Plinetype_cont   linetype_cont;      /* linetype continuity         */
Phighl_method    highl_method;       /* highlighting method         */
Pint_list        active_textures;      /* active textures           */
Pint_list        back_active_textures; /* back active textures      */
Pperspect_corr   perspect_corr;   /* textr perspective correction   */
struct Ptexture_res_opt_heur {
   Pint          opt_hint;          /* optimization hint           */
   Pint_list     usage_priorities; /* texture usage priorities     */
} texture_res_opt_heur;
Pint_list        alpha_src_sel;   /* alpha source selector         */
struct Pcond_exec_struct {
  Pint           struct_id;          /* structure identifier        */
  Ptest          test;               /* condition test              */
} cond_exec_struct;
struct Pcond_inst_struct {
  Pint           struct_id;          /* structure identifier        */
  Ptest          test;               /* condition test              */
} cond_inst_struct;
Ptest            cond_return;        /* condition test              */
struct Pcond_skip_elems {
  Pint           skip_count;         /* number of elements to skip  */
  Ptest          test;               /* condition test              */
} cond_skip_elems;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  struct Pcond_skip_to_label {
   Pint           label;              /* label to which to skip      */
   Ptest          test;               /* condition test              */
  } cond_skip_to_label;
  /* end of Full PHIGS element data */

} Pelem_data;
```

Pelem_type **ELEMENT TYPE**

```
typedef enum {
   /* start of PHIGS enumerations */
   ...,
   /* end of PHIGS enumerations */
   /* start of PHIGS PLUS enumerations */
   ...,
   /* end of PHIGS PLUS enumerations */
   /* start of Full PHIGS enumerations */
   PELEM_APPL_INT,
   PELEM_APPL_REAL,
   PELEM_CIRCLE3,
   PELEM_CIRCLE,
   PELEM_CIRCULAR_ARC3,
   PELEM_CIRCULAR_ARC,
   PELEM_ELLIPSE3,
   PELEM_ELLIPSE,
   PELEM_ELLIPTICAL_ARC3,
   PELEM_ELLIPTICAL_ARC,
   PELEM_FILL_CIRCLE3,
   PELEM_FILL_CIRCLE,
   PELEM_CIRCULAR_ARC_CLOSE3,
   PELEM_CIRCULAR_ARC_CLOSE,
   PELEM_FILL_ELLIPSE3,
   PELEM_FILL_ELLIPSE,
   PELEM_ELLIPTICAL_ARC_CLOSE3,
   PELEM_ELLIPTICAL_ARC_CLOSE,
   PELEM_HIGHL_IND,
   PELEM_LINETYPE_ADAPT,
   PELEM_LINETYPE_CONT,
   PELEM_LINETYPE_OFFSET,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    PELEM_HIGHL_METHOD,
    PELEM_ACTIVE_TEXTURES,
    PELEM_BACK_ACTIVE_TEXTURES,
    PELEM_TEXTURE_PERSPECT_CORR,
    PELEM_TEXTURE_SAMPLING_FREQ,
    PELEM_TEXTURE_RES_OPT_HEUR,
    PELEM_TRANSPARENCY,
    PELEM_BACK_TRANSPARENCY,
    PELEM_ALPHA_SRC_SEL,
    PELEM_ALPHA_DATA_SEL_IND,
    PELEM_INST_STRUCT,
    PELEM_COND_EXEC_STRUCT,
    PELEM_COND_INST_STRUCT,
    PELEM_COND_RETURN,
    PELEM_COND_SKIP_ELEMS,
    PELEM_COND_SKIP_TO_LABEL,
    PELEM_PUSH_ST,
    PELEM_POP_ST
    /* end of Full PHIGS enumerations */

} Pelem_type;
```

`Pin_class` **INPUT CLASS**

```
typedef enum {

    /* start of Basic PHIGS enumerations */
    PIN_NONE,
    PIN_LOC,
    PIN_STROKE,
    PIN_VAL,
    PIN_CHOICE,
    PIN_PICK,
    PIN_STRING,
    /* end of Basic PHIGS enumerations */
    /* start of Full PHIGS enumerations */
    PIN_SET,
    PIN_COMPOSITE
    /* end of Full PHIGS enumerations */

} Pin_class;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pint_style` **INTERIOR STYLE**

```
typedef enum {
    /* start of Basic PHIGS enumerations */
    ...
    /* end of Basic PHIGS enumerations */
    /* start of Full PHIGS enumerations */
    PSTYLE_TEXTURE
    /* end of Full PHIGS PLUS enumerations */
} Pint_style;
```

`Popen_struct_status` **OPEN STRUCTURE STATUS**

```
typedef enum {
    /* start of Basic PHIGS enumerations */
    PSTRUCT_NONE,
    PSTRUCT_OPEN,
    /* end of Basic PHIGS enumerations */
    /* start of Full PHIGS enumerations */
    PSTRUCT_DI_OPEN
    /* end of Full PHIGS enumerations */
} Popen_struct_status;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

11.3 Modifications to *PHIGS PLUS profile* type definitions

`Pcolr_rep_ptr` **COLOUR REPRESENTATION POINTER**

```
typedef union {
   /* start of PHIGS PLUS union members */
   Prgb       *rgb;          /* pointer to RGB colour values */
   Pcieluv    *cieluv;       /* pointer to CIELUV pointer values */
   Phls       *hls;          /* pointer to HLS pointer values */
   Phsv       *hsv;          /* pointer to HSV pointer values */
   /* end of PHIGS PLUS union members */
   /* start of Full PHIGS union members */
   Prgba      *rgba;         /* pointer to RGBA colour values */
   Pcieluva   *cieluva;      /* pointer to CIELUVA pointer values */
   Phlsa      *hlsa;         /* pointer to HLSA pointer values */
   Phsva      *hsva;         /* pointer to HSVA pointer values */
   /* end of Full PHIGS union members */
   ...                       /* implementation defined */
} Pcolr_rep_ptr;
```

`Ppat_rep_plus` **PATTERN REPRESENTATION PLUS**

```
typedef Pgcolr_array Ppat_rep_plus;
```

11.4 Implementation dependent *Full PHIGS profile* type definitions

---

`Pbackg_method` **POSTING GROUP BACKGROUND METHOD**

```
typedef struct {
   Pint                 method;        /* background method          */
   union Pbackg_method_data {
      struct Pcolr_table_0 {
         Pint      unused;             /* unused                     */
      } colr_table_0;
      struct Pcolr_ind {
         Pint      colr_ind;           /* colour index               */
      } colr_ind;
      struct Pcolr {
         Pcolr_rep colr_rep;           /* target address             */
      } colr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Pimage_resource {
         Pint      image_resource_id; /* image resource identifier   */
      } image_resource;
      ...                             /* implementation defined      */
   } data;

} Pbackg_method;
```

`Pclamp_method` **BOUNDARY CLAMP METHOD DATA RECORD**

```
typedef struct {
   Pint           method;          /* clamp method                   */
   union Pclamp_method_data {
      Pint        unused;          /* used by method 1               */
      Pgcolr      clamp_colr;      /* used by method 2               */
      ...                          /* implementation defined         */
   } data;

} Pclamp_method;
```

`Pcoord_src` **COOORDINATE SOURCE DATA RECORD**

```
typedef struct {
   Pint           coord_src;       /* coordinate source              */
   union Pcoord_src_data {         /* coordinate source data record  */
      Pint        unused;          /* used by coord sources 1 - 5    */
      Pint        coord_ind;       /* used by coord source 6         */
      Pmatrix3    refl_matrix;     /* used by coord sources 7 and 8  */
      ...                          /* implementation defined         */
   } data;

} Pcoord_src;
```

`Pdi_pick_data` **DIRECT INTERPRETATION PICK DATA RECORD**

```
typedef struct {
   union Pdi_pick_pets {
      struct Pdi_pick_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Pdi_pick_pet_r1 {
         . . .                              /* impl. dependent              */
      } pet_r1;  /* For PET 1 */
      ...                                   /* implementation defined PET's*/
   } pets;
   union Pdi_pick_measure_data {
      struct Pdi_pick_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      . . .        /* data for impl. defined measure processes */
   } measure_data;
   union Pdi_pick_trigger_data {
      struct Pdi_pick_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .        /* data for impl. defined di_pick processes */
   } trigger_data;
   union Pdi_pick_ack_data {
      struct Pdi_pick_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      . . .        /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pdi_pick_data;
```

Pdi_pick_data3 **DIRECT INTERPRETATION PICK DATA RECORD 3**

```
typedef struct {
   union Pdi_pick3_pets {
      struct Pdi_pick3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pdi_pick3_pet_r1 {
         . . .                              /* impl. dependent              */
      } pet_r1;  /* For PET 1 */
      . . .        /* data for impl. defined pets */
   } pets;
   union Pdi_pick3_measure_data {
      struct Pdi_pick3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      . . .          /* data for impl. defined measure processes */
   } measure_data;
   union Pdi_pick3_trigger_data {
      struct Pdi_pick3_trig_other {
         Pint        unused;
      } trig_other; /* When no trigger-specific data is required */
      . . .          /* data for impl. defined di_pick processes */
   } trigger_data;
   union Pdi_pick3_ack_data {
      struct Pdi_pick3_ack_other {
         Pint        unused;
      } ack_other; /* When no ack.-specific data is required */
      . . .          /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pdi_pick_data3;
```

Phighl_method **HIGHLIGHTING METHOD**

```
typedef struct {
   Pint                 method;
   union _Phighl_data {
      Pint          unused;
      Pfloat        blink_rate;
      Pint          colr_ind;
      Pgcolr        colr;
      struct _Phighl_blink_colr {
         Pfloat     blink_rate;
         Pgcolr     blink_colr;
      } highl_blink_colr;
      ...           /* implementation defined methods */
   } highl_data;
} Phighl_method;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pimage_res **IMAGE RESOURCE**

```
typedef struct {
   Pint image_spec_method;
   union _Pimage_data {
      struct Puncomp_colr_ind_array {
         Pint      *colr_ind_array;    /* uncompressed array of       */
                                       /* colour indices              */
      } uncomp_colr_ind_array;
      struct Puncomp_colr_array {
         Pgcolr_array  *colr_array;    /* uncompressed array of       */
                                       /*  colours                    */
      } uncomp_colr_array;
      struct Pwindow_sys_bitmap {
         Pint          handle;         /* bitmap handle              */
      } window_sys_bitmap;
      struct Pvideo_sig_chan_id {
         Pint          chan_id;        /* video signal channel id    */
      } video_sig_chan_id;
      struct Pluminance {
         Pfloat_array  *lum_values;    /* uncompressed array of*/
                                       /* luminance values*/
      } luminance;
      struct Pluminance_alpha {
         Plum_alpha_array  *lum_values; /* uncompressed array of     */
                                        /* luminance values          */
      } luminance_alpha;
      struct Pmipmap_colrs {
         Pgcolr_array_set  colr_arrays; /* set of uncompressed       */
                                        /* colour arrays             */
      } mipmap_colrs;
      struct Pmipmap_luminance {
         Pfloat_array_set  lum_values;  /* set of uncompressed       */
                                        /* luminance arrays          */
      } mipmap_luminance;
      struct Pmipmap_luminance_alpha {
         Plum_alpha_array_set lum_values;   /* set of uncompressed   */
                                            /* luminance/alpha arrays*/
      } mipmap_luminance_alpha;
      ...          /* implementation defined methods */
   } image_data;
} Pimage_res;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pin_class_data **INPUT CLASS DATA RECORD**

```
typedef union _Pin_class_data {

   Pint                unused;
   Ploc_data           loc;
   Ploc_data3          loc3;
   Pstroke_data        stroke;
   Pstroke_data3       stroke3;
   Pval_data           val;
   Pval_data3          val3;
   Pchoice_data        choice;
   Pchoice_data3       choice3;
   Ppick_data          pick;
   Ppick_data3         pick3;
   Pstring_data        string;
   Pstring_data3       string3;
   Pset_data           set;
   Pcomposite_data     composite;
   ...                                  /* other classes of input data */

} Pin_class_data;
```

Pin_class_measure **INPUT CLASS MEASURE**

```
typedef union _Pin_class_measure {

   struct {
      Pint          view_ind;
      Ppoint        loc_pos;
   } loc;

   struct {
      Pint          view_ind;
      Ppoint3       loc_pos;
   } loc3;

   struct {
      Pint          view_ind;
      Ppoint_list   points;
   } stroke;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   struct {
      Pint          view_ind;
      Ppoint_list3  points;
   } stroke3;

   Pfloat               val;

   struct {
      Pin_status    istat;
      Pint          selection;
   } choice;

   struct {
      Pin_status    istat;
      Ppick_path    *path;
   } pick;

   char                 *string;

   Pset_measure                 *set;

   Pcomposite_measure           *composite;

   ...                  /* other classes which may be added */

} Pin_class_measure;
```

Pmarker_data **MARKER DATA**

```
typedef struct {

   Pint                 shape_type;     /* shape type                    */

   union _Pshape_data {
      Ppoint_list polyline;            /* for PSHAPE_POLYLINE           */
      Ppoint_list fill_area;           /* for PSHAPE_FILL_AREA          */
      Ppoint_list convex_fill_area;    /* for PSHAPE_CONVEX_FILL_AREA   */
      ...                              /* implementation defined        */
   } shape_data;                       /* marker shape data             */

} Pmarker_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Ppattern **PATTERN**

```
typedef struct {
   Pint                 type;          /* pattern type */
   union _Ppattern_data {
      struct Ppat_colr_ind_array {
         Ppat_rep  pat_rep;            /* pattern representation      */
      } pat_colr_ind_array;
      struct Ppat_colr_array {
         Ppat_rep_pluspat_rep_plus;    /* pattern representation plus */
      } pat_colr_array;
      struct Ppat_image_res {
         Pint      image_res_id;       /* pattern image resource      */
      } pat_image_res;
      ...                              /* implementation defined      */
   } data;                             /* pattern data */
} Ppattern;
```

Ptarg_op **TARGET OPERATION**

```
typedef struct {
   Pint                 type;          /* target operation type       */
   union _Ptarget_op_data {
      struct Prend_targ {
         Ptarg_addrtarg_addr;         /* target address              */
      } rend_targ;
      struct Pdisp_targ {
         Ptarg_addrtarg_addr;         /* target address              */
      } disp_targ;
      struct Pclear_targ {
         Ptarg_addrtarg_addr;         /* target address              */
      } clear_targ;
      struct Pcopy_targ {
         Ptarg_addrsrc_targ_addr;     /* source target address       */
         Ptarg_addrdest_targ_addr;    /* destination target address  */
      } copy_targ;
      ...                             /* implementation defined      */
   } data;                            /* target operation data       */
} Ptarg_op;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Ptest **TEST**

```
typedef struct {
   Pint                 method;          /* test method                  */
   union _Ptest_data {
      struct Ptest_always_fail {
         Pint               unused;
      } test_always_fail;
      struct Ptest_always_succeed {
         Pint               unused;
      } test_always_succeed;
      struct Ptest_extent_3 {
         Plimit3            rect;          /* extent rectangle          */
         Ptest_comp         relation;      /* test relation             */
         Pfloat             threshhold;    /* threshhold value          */
      } test_extent_3;
      struct Ptest_extent {
         Plimit             rect;          /* extent rectangle          */
         Ptest_comp         relation;      /* test relation             */
         Pfloat             threshhold;    /* threshhold value          */
      } test_extent;
      struct Ptest_bounds_3 {
         Ppoint_list_list3 point_lists;    /* test object               */
         Pclip_cond         cond;          /* test condition            */
      } test_bounds_3;
      struct Ptest_bounds {
         Ppoint_list_list  point_lists;    /* test object               */
         Pclip_cond         cond;          /* test condition            */
      } test_bounds;
      struct Ptest_nameset {
         Pint               filter_type;   /* type of filter            */
         Pint               filter_id;     /* filter identifier         */
         Pfilter_op         cond;          /* test condition            */
      } test_nameset;
      struct Ptest_appl_int {
         Pint               appl_id;       /* appl. integer selector    */
         Ptest_comp_rel     comparison;    /* test comparison           */
         Pint               data;          /* appl integer data value   */
      } test_appl_int;
      struct Ptest_appl_real {
         Pint               appl_id;       /* appl. real selector       */
         Ptest_comp_rel     comparison;    /* test comparison           */
         Pfloat             data;          /* appl real data value      */
      } test_appl_real;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Ptest_cond_flags {
         Pcond_flags_eval_op  op;      /* condition flags eval. op.   */
      } test_cond_flags;
      struct Ptest_appl_int_as_logical {
         Pint                 appl_id;     /* appl. integer selector   */
         Ptest_comp_logical   comparison;  /* test comparison          */
         Pint                 data;        /* appl integer data value  */
      } test_appl_int;
      ...                                  /* implementation defined   */
   } test_data;       /*                   test data record*/
} Ptest;
```

Ptexture_compos **TEXTURE COMPOSITION DATA RECORD**

```
typedef struct {
   Pint           method;          /* composition method              */
   union {
      Pint        unused;          /* used by methods 1, 2, and 4     */
      struct {                     /* used by method 3                */
        Pgcolr    environ;         /* environment colour              */
        Pint      gamma_chan_sel;  /* gamma channel selector          */
      } blend_env;
      Pgcolr      background_colr; /* used by method 5                */
      ...                          /* implementation defined          */
   } data;
} Ptexture_compos;
```

Ptrav_res_op **TRAVERSAL RESOURCE OPERATION**

```
typedef struct {
   Pint                type;          /* trav. res. operation type   */
   union Ptrav_res_op_data {
      struct Ptrav_res_clear {
         Pint      res_id;            /* traversal resource id.      */
      } trav_res_clear;
      struct Ptrav_res_copy {
         Pint      src_res_id;        /* source traversal res. id.   */
         Pint      dest_res_id;       /* destination trav. res. id.  */
      } trav_res_copy;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
        ...                                 /* implementation defined      */
    } data;                                 /* trav. res. operation data   */
} Ptrav_res_op;
```

11.5 Implementation independent type definitions

`Paccess_flag` **ACCESS FLAG**

```
typedef enum {
    PACCESS_REFERENCED = 0,
    PACCESS_COPIED = 1
} Paccess_flag;
```

`Pack_process` **ACKNOWLEDGEMENT PROCESS**

```
typedef struct {
    Pack_type           type;
    Pint                process_id;
} Pack_process;
```

`Pack_process_list` **ACKNOWLEDGEMENT PROCESS LIST**

```
typedef struct {
    Pint                count;
    Pack_process        *procs;
} Pack_process_list;
```

`Pack_type` **ACKNOWLEDGEMENT TYPE**

```
typedef enum {
    P_ACK_ACCEPTANCE = 0,
    P_ACK_NONACCEPTANCE = 1
} Packnowledgement_type;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Palpha_facs **ALPHA FACILITIES**

```
typedef struct {

   Pint_list  alpha_srcs;      /* list available alpha sources       */
   Pint_list  transp_modes;    /* list transparency modes supported  */

} Palpha_facs;
```

Passoc_flag **ASSOCIATION FIAG**

```
typedef enum {

   PNOTASSOCIATED = 0,
   PASSOCIATED = 1

} Passoc_flag;
```

Patomic_lid_facs **ATOMIC LOGICAL INPUT DEVICE FACILITIES**

```
typedef struct {

   Pint      max_simul_trig,   /* max simul. trigger procs          */
   Pint      max_simul_echo,   /* max simul. echo procs             */
   Pint      max_simul_ack,    /* max simul. acknowledgement procs  */
   Pint_list meas_proc_ids,    /* available measure process ids     */
   Pint_list trig_proc_ids,    /* available trigger process ids     */
   Pint_list echo_proc_ids,    /* available echo process ids        */
   Pint_list ack_proc_ids      /* available acknowledgement process ids
                                  */

} Patomic_lid_facs;
```

Pbackg_redisplay **BACKGROUND REDISPLAY FLAG**

```
typedef enum {

   PBACKG_REDISPLAY_OFF = 0,
   PBACKG_REDISPLAY_ON = 1

} Pbackg_redisplay;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pbackg_style **BACKGROUND STYLE**

```
typedef enum {
    PBACKG_STYLE_TRANSPARENT = 0,
    PBACKG_STYLE_OPAQUE = 1
} Pbackg_style;
```

Pboolean **BOOLEAN**

```
typedef enum {
    PFALSE = 0,
    PTRUE = 1
} Pboolean;
```

Pborder_indic **BORDER INDICATOR**

```
typedef enum {
    PBORDER_OFF = 0,
    PBORDER_ON = 1
} Pborder_indic;
```

Pcieluva **CIE L* U* V* WITH ALPHA**

```
typedef struct {
    Pfloat      cieluv_x;       /* x coefficient          */
    Pfloat      cieluv_y;       /* y coefficient          */
    Pfloat      cieluv_y_lum;   /* y luminance            */
    Pfloat      alpha;          /* alpha                  */
} Pcieluva;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pclip_cond **CLIP CONDITION**

```
typedef enum {
  PCLIPCOND_IS_NOT_CLIPPED = 0,
  PCLIPCOND_IS_PARTIALLY_CLIPPED = 1,
  PCLIPCOND_IS_FULLY_CLIPPED = 2
} Pclip_cond;
```

Pclosure **CLOSURE TYPE**

```
typedef enum {
   PCLOSURE_PIE = 0,
   PCLOSURE_SEGMENT = 1
} Pclosure;
```

Pcomposite_data **COMPOSITE DATA**

```
typedef struct {
   Pint                  count;
   Pcomposite_individual *individual;
} Pcomposite_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pcomposite_facs **COMPOSITE FACILITIES**

```
typedef struct {

    Pint      max_simul_meas,        /* max simul. measure procs       */
    Pint      max_simul_trig,        /* max simul. trigger procs       */
    Pint      max_simul_echo,        /* max simul. echo procs          */
    Pint      max_simul_ack,         /* max simul. acknowledgement procs
                                        */
    Pint_list loc_meas_proc_ids,     /* avail. locator measure proc. ids
                                        */
    Pint_list stroke_meas_proc_ids, /* avail. stroke measure proc. ids
                                        */
    Pint_list val_meas_proc_ids,     /* avail. valuator measure proc. ids
                                        */
    Pint_list choice_meas_proc_ids, /* avail. choice measure proc. ids
                                        */
    Pint_list pick_meas_proc_ids,    /* avail. pick measure proc. ids  */
    Pint_list string_meas_proc_ids, /* avail. string measure proc. ids
                                        */
    Pint_list set_meas_proc_ids,     /* avail. set measure proc. ids   */
    Pint_list composite_meas_proc_ids,/*avail. composite meas. proc. ids
                                        */
    Pint_list trig_proc_ids,         /* available trigger process ids  */
    Pint_list loc_echo_proc_ids,     /* available locator echo proc. ids
                                        */
    Pint_list stroke_echo_proc_ids, /* available stroke echo proc. ids
                                        */
    Pint_list val_echo_proc_ids,     /* available valuator echo proc. ids
                                        */
    Pint_list choice_echo_proc_ids, /* available choice echo proc. ids
                                        */
    Pint_list pick_echo_proc_ids,    /* available pick echo proc. ids  */
    Pint_list string_echo_proc_ids, /* available string echo proc. ids
                                        */
    Pint_list set_echo_proc_ids,     /* available set echo proc. ids   */
    Pint_list comp_echo_proc_ids,    /* available composite echo proc. ids
                                        */
    Pint_list ack_proc_ids           /* available ack. proc. ids       */

} Pcomposite_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pcomposite_individual **COMPOSITE INDIVIDUAL**

```
typedef struct _Pcomposite_individual {

   Pin_class        class;
   _Pin_class_data  *data;

} Pcomposite_individual;
```

Pcomposite_measure **COMPOSITE MEASURE**

```
typedef struct _Pcomposite_measure {

   Pint         count;                /* number of measure components*/
   _Pmeasure    *measures;            /* list of measure components  */

} Pcomposite_measure;
```

Pcond_trav_facs **CONDITIONAL TRAVERSAL FACILITIES**

```
typedef struct {

  Pint      num_invis_filters;/*number of avail. invisibility filters
                               */
  Pint      num_highl_filters;/*number of avail. highlighting filters
                               */
  Pint      num_pick_filters; /*number of avail. pick filters       */
  Pint      num_appl_filters; /*number of avail. application filters
                               */
  Pint      num_appl_integers;/*number of application integer       */
                           ./*  values supported                    */
  Pint      num_appl_reals;  /*  number of application real values  */
                             /*  supported                          */
  Pint_list test_methods;    /*  list of test methods available     */

} Pcond_trav_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pcond_flags_eval_op **CONDITION FLAGS EVALUATION OPERATOR**

```
typedef enum {
    PCOND_ALL_ENABLED,
    PCOND_ALL_DISABLED,
    PCOND_NOT_ALL_DISABLED,
    PCOND_NOT_ALL_ENABLED,
    PCOND_ALWAYS_PASS,
    PCOND_NEVER_PASS
} Pcond_flags_eval_op;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pcond_flag_mask **CONDITION FLAG MASK**

```
typedef union {
   unsigned int mask;
   struct _Pcond_flag_set {
      unsigned int flag0:1;
      unsigned int flag1:1;
      unsigned int flag2:1;
      unsigned int flag3:1;
      unsigned int flag4:1;
      unsigned int flag5:1;
      unsigned int flag6:1;
      unsigned int flag7:1;
      unsigned int flag8:1;
      unsigned int flag9:1;
      unsigned int flag10:1;
      unsigned int flag11:1;
      unsigned int flag12:1;
      unsigned int flag13:1;
      unsigned int flag14:1;
      unsigned int flag15:1;
      unsigned int flag16:1;
      unsigned int flag17:1;
      unsigned int flag18:1;
      unsigned int flag19:1;
      unsigned int flag20:1;
      unsigned int flag21:1;
      unsigned int flag22:1;
      unsigned int flag23:1;
      unsigned int flag24:1;
      unsigned int flag25:1;
      unsigned int flag26:1;
      unsigned int flag27:1;
      unsigned int flag28:1;
      unsigned int flag29:1;
      unsigned int flag30:1;
      unsigned int flag31:1;
   } flags;
} Pcond_flag_mask;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pcond_flags **CONDITION FLAGS**

```
typedef struct {

   Pcond_flag_mask    enable_mask;   /* condition flag enable mask  */
   Pcond_flag_mask    disable_mask;  /* condition flag disable mask */

} Pcond_flags;
```

Pcond_flag_test **CONDITION FLAG TEST**

```
typedef struct {

   Ptest              test;          /* test operation              */
   Pcond_flags        mask;          /* condition test mask         */

} Pcond_flag_test;
```

Pcond_test_list **CONDITION TEST LIST**

```
typedef struct {

   Pint               num_tests;     /* number of condition test    */
   Pcond_flag_test    *tests;        /* list of condition flag tests*/

} Pcond_test_list;
```

Pconfig_name **CONFIGURATION NAME**

```
typedef struct {

   Pint            name_size;       /* size of name                */
   char            *name;           /* configuration setting name  */

} Pconfig_name;
```

Pconfig_name_list **CONFIGURATION NAME LIST**

```
typedef struct {

   Pint            list_size;       /* number of names             */
   Pconfig_name    *name;           /* configuration name list     */

} Pconfig_name_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pconfig_setting_facs` **CONFIGURATION SETTING FACILITIES**

```
typedef struct {
    Pint          num_supported; /* number of simultaneously supported
                                    */
                                 /* device coordinate clip regions    */
} Pconfig_setting_facs;
```

`Pconfig_settings` **CONFIGURATION SETTINGS**

```
typedef struct {
    char             *name;          /* configuration setting name    */
    Pconfig_value    *value;         /* configuration setting value   */
} Pconfig_settings;
```

`Pconfig_value` **CONFIGURATION VALUE**

```
typedef union {
    Pint         int_value;          /* integer configuration value   */
    Pfloat       real_value;         /* real configuration value      */
    char         *string_value;      /* string configuration value    */
} Pconfig_value;
```

`Pdc_clip_facs` **DEVICE COORDINATE CLIP REGION FACILITIES**

```
typedef struct {
    Pint           num_supported;   /* number of simultaneously sup-
                                       ported*/
                                    /* device coordinate clip regions */
} Pdc_clip_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pdepth_cue_mask` **DEPTH CUE MASK**

```
typedef struct {
   unsigned int mode:1;
   unsigned int ref_planes:1;
   unsigned int scale:1;
   unsigned int gcolr:1;
   unsigned int :28;
} Pdepth_cue_mask;
```

`Pdi_mode` **DIRECT INTERPRETATION MODE**

```
typedef enum {
   PDIM_INTERPRET_NONE = 0,
   PDIM_INTERPRET_STATE = 1,
   PDIM_INTERPRET_ALL = 2
} Pdi_mode;
```

`Pdi_trav_facs` **DIRECT INTERPRETATION TRAVERSAL FACILITIES**

```
typedef struct {
   Pint     num_trav_procs;         /* number of simultaneously posted
                                       */
                                    /* traversal processes           */
   Pint     num_rend_algorithms;    /* number of multi-pass algorithms
                                       */
                                    /* supported for rendering       */
   Pint     num_pick_algorithms;    /* number of multi-pass algortihms
                                       */
                                    /* supported for picking         */
} Pdi_trav_facs;
```

`Pdyns_post_grps` **DYNAMICS OF POSTING GROUPS**

```
typedef struct {
   Pdyn_mod          posting_status;/* posting status modification */
} Pdyns_post_grps;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pdyns_ws_attrs_texture` **DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE**

```
typedef struct {
   Pdyn_mod   texture_rep;          /* texture representation          */
} Pdyns_ws_attrs_texture;
```

`Pecho_measure_proc` **ECHO MEASURE PROCESS**

```
typedef struct {
   Pin_class          dev_class;   /* input class                     */
   Pint               instance;    /* i = ith of class in LID def.    */
} Pecho_measure_proc;
```

`Pecho_process` **ECHO PROCESS**

```
typedef struct {
   Pint               pet;
   Pecho_measure_proc *measures_echoed;
   Pint               process_id;
} Pecho_process;
```

`Pecho_process_list` **ECHO PROCESS LIST**

```
typedef struct {
   Pint               count;
   Pecho_process      *procs;
} Pecho_process_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pedge_bundle_full **EDGE BUNDLE FULL**

```
typedef struct {
   Pedge_flag          flag;          /* edge flag                    */
   Pint                type;          /* edgetype                     */
   Pfloat              width;         /* edgewidth scale factor       */
   Pgolr               colr;          /* edge colour                  */
   Plinetype_adapt     adapt;         /* edgetype adaptability        */
   Plinetype_cont      cont;          /* edgetype continuity          */
   Pfloat              offset;        /* edgetype offset              */
   Plinecap            cap;           /* edgecap                      */
   Plinejoin           join;          /* edgejoin                     */
   Pfloat              limit;         /* edgemitre limit              */
} Pedge_bundle_full;
```

Pedge_mask **EDGE MASK**

```
typedef struct {
   unsigned int flag:1;
   unsigned int type:1;
   unsigned int width:1;
   unsigned int colr:1;
   unsigned int adapt:1;
   unsigned int cont:1;
   unsigned int offset:1;
   unsigned int cap:1;
   unsigned int join:1;
   unsigned int limit:1;
   unsigned int :22;
} Pedge_mask;
```

Pedgetype_adapt **EDGETYPE ADAPTABILITY**

```
typedef enum {
   PEDTA_EXACT = 0,
   PEDTA_ADAPT = 1
} Pedgetype_adapt;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pedgecap **EDGECAP**

```
typedef enum {
    PEDC_BUTT = 0,
    PEDC_ROUND = 1,
    PEDC_SQUARE = 2
} Pedgecap;
```

Pedgetype_cont **EDGETYPE CONTINUITY**

```
typedef enum {
    PEDTC_CONTINUOUS = 0,
    PEDTC_RESTART = 1
} Pedgetype_cont;
```

Pedgejoin **EDGEJOIN**

```
typedef enum {
    PEDJ_FLAT = 0,
    PEDJ_MITRE = 1,
    PEDJ_ROUND = 2,
    PEDJ_BEVEL = 3
} Pedgejoin;
```

Pext_pat_facs **EXTENDED PATTERN FACILITIES**

```
typedef struct {
    Pint_list pattern_types;      /* list of supported pattern types   */
    Pint_list image_spec_methods;/* list of supported image specifcation
                                    */
                                  /* methods                           */
} Pext_pat_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pfilter_op` **FILTER RELATIONS**

```
typedef enum {
    PFILTER_FAILS = 0,
    PFILTER_PASSES = 1
} Pfilter_op;
```

`Pfloat_array` **FLOAT ARRAY**

```
typedef struct {
    Pint_size       dims;           /* float array dimensions      */
    Pfloat          *floats;        /* float array                 */
} Pfloat_array;
```

`Pfloat_array_set` **FLOAT ARRAY SET**

```
typedef struct {
    Pint            num_arrays;     /* number of float arrays in set  */
    Pfloat_array    *floats;        /* float arrays                   */
} Pfloat_array_set;
```

`Pgcolr_array` **GENERAL COLOUR ARRAY**

```
typedef struct {
    Pint            colr_type;      /* colour type                 */
    Pcolrv_array    colrs;          /* colour array                */
} Pgcolr_array;
```

`Pgcolr_array_set` **GENERAL COLOUR ARRAY SET**

```
typedef struct {
    Pint            colr_type;      /* colour type                 */
    Pint            num_arrays;     /* number of arrays in set     */
    Pcolrv_array    *colrs;         /* set of colour arrays        */
} Pgcolr_array_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Phighl_facs` **HIGHLIGHTING FACILITIES**

```
typedef struct {

   Pint           num_pred;        /* number of predefined methods   */
   Pint_list      methods;         /* number of highl. methods supp. */
   Pint           max_highl_ind;   /* max highl. table indices       */

} Phighl_facs;
```

`Phlsa` **HUE LIGHTNESS SATURATION WITH ALPHA**

```
typedef struct {

   Pfloat     hue;             /* hue                                 */
   Pfloat     lightness;       /* lightness                           */
   Pfloat     saturation;      /* saturation                          */
   Pfloat     alpha;           /* alpha                               */

} Phlsa;
```

`Phsva` **HUE SATURATION VALUE WITH ALPHA**

```
typedef struct {

   Pfloat     hue;             /* hue                                 */
   Pfloat     saturation;      /* saturation                          */
   Pfloat     value;           /* value                               */
   Pfloat     alpha;           /* alpha                               */

} Phsva;
```

`Pimage_res_facs` **IMAGE RESOURCE FACILITIES**

```
typedef struct {

   Pint_list image_spec_methods;/* list of supported image           */
                                /* specification methods             */
   Pint      num_def_ids;       /* number of definable image resource
                                   */
                                /* identifiers                       */
   Pint      num_predef_ids;    /* number of predefined image resources
                                   */

} Pimage_res_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

_Pin_class_data **INPUT CLASS DATA RECORD**

```
union _Pin_class_data;   /* used for defining Pset_data and
                            Pcomposite_data; */
```

Pin_class_measure **INPUT CLASS MEASURE**

```
union _Pin_class_measure;   /* used for defining Pmeasure and
                               Pset_measure */
```

Pinterior_mask **INTERIOR MASK**

```
typedef struct {

   unsigned int style:1;
   unsigned int style_ind:1;
   unsigned int colr:1;
   unsigned int shad_method:1;
   unsigned int :28;

} Pinterior_mask;
```

Plight_attach **LIGHT SOURCE ATTACHMENT**

```
typedef struct {

   Pint     light_ws_id;              /* workstation id for view     */
   Pint     light_src_ind;            /* view index                  */
   Pint     light_action_type;        /* view action type            */
   Pint     transfer_type;            /* measure transfer type       */

} Plight_attach;
```

Plimit_list **LIMIT LIST**

```
typedef struct {

   Pint              num_limits;     /* number of limits            */
   Plimit            *limits;        /* list of limits              */

} Plimit_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Plimit3_list **LIMIT 3 LIST**

```
typedef struct {
    Pint                num_limits;     /* number of limits          */
    Plimit3             *limits;        /* list of limits            */
} Plimit3_list;
```

Pline_bundle_full **POLYLINE BUNDLE FULL**

```
typedef struct {
    Pint                type;           /* linetype                  */
    Pfloat              width;          /* linewidth scale factor    */
    Pgolr               colr;           /* polyline colour           */
    Pint                shad_method;    /* polyline shading method   */
    Pcurve_approx_crit_data
                        curve_approx_crit_data;
                                        /* curve approx. criteria    */
    Plinetype_adapt     adapt;          /* linetype adaptability     */
    Plinetype_cont      cont;           /* linetype continuity       */
    Pfloat              offset;         /* linetype offset           */
    Plinecap            cap;            /* linecap                   */
    Plinejoin           join;           /* linejoin                  */
    Pfloat              limit;          /* linemitre limit           */
} Pline_bundle_full;
```

Plinecap **LINECAP**

```
typedef enum {
    PLNC_BUTT = 0,
    PLNC_ROUND = 1,
    PLNC_SQUARE = 2
} Plinecap;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Plinejoin **LINEJOIN**

```
typedef enum {
    PLNJ_FLAT = 0,
    PLNJ_MITRE = 1,
    PLNJ_ROUND = 2,
    PLNJ_BEVEL = 3
} Plinejoin;
```

Plinetype_adapt **LINETYPE ADAPTABILITY**

```
typedef enum {
    PLNTA_EXACT = 0,
    PLNTA_ADAPT = 1
} Plinetype_adapt;
```

Plinetype_cont **LINETYPE CONTINUITY**

```
typedef enum {
    PLNTC_CONTINUOUS = 0,
    PLNTC_RESTART = 1
} Plinetype_cont;
```

Plinetype_def_facs **LINETYPE DEFINITION FACILITIES**

```
typedef struct {
    Pint      num_definable;       /* number definable linetypes      */
    Pint      num_predefined;      /* number predefined linetypes     */
    Pint      max_dash_segments;   /* maximum dash segments supported */
    Pfloat    offset;              /* offset to definable linetypes   */
    Pint_list predef_def_linetypes /* list of predef. definable linetypes
                                     */
} Plinetype_def_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Plum_alpha_array` **LUMINANCE ALPHA ARRAY**

```
typedef struct {

   Pint_size        dims;        /* array dimensions                    */
   Plum_alpha_pair  *values;     /* 2D array of luminance/alpha pairs */

} Plum_alpha_array;
```

`Plum_alpha_array_set` **LUMINANCE ALPHA ARRAY SET**

```
typedef struct {

   Pint             num_arrays; /* number of arrays in set          */
   Plum_alpha_array *sets;      /* set of 2D luminance/alpha arrays */

} Plum_alpha_array_set;
```

`Plum_alpha_pair` **LUMINANCE ALPHA PAIR**

```
typedef struct {

   Pfloat          lum_value;          /* luminance_value            */
   Pfloat          alpha_value;        /* alpha value                */

} Plum_alpha_pair;
```

`Pmarker_desc` **MARKER DESCRIPTOR**

```
typedef struct {

   Pint             number_shapes;    /* number of shapes           */
   Pmarker_data     *marker;          /* array of shape descriptors */

} Pmarker_desc;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pmarker_type_def_facs **MARKER TYPE DEFINITION FACILITIES**

```
typedef struct {

   Pint       num_definable;       /* number definable marker types  */
   Pint       num_predefined;      /* number predefined marker types */
   Pfloat     offset;              /* offset to definable linetypes  */
   Pint       max_allowable_pts;   /* maximum allowable points       */
   Pint_list  shape_types;         /* list of available shape types  */
   Pint_list  predef_marker_types  /* list of predefined marker types
                                      */

} Pmarker_type_def_facs;
```

Pmeasure **MEASURE**

```
typedef struct _Pmeasure {

   Pin_class          class;
   _Pin_class_measure *data;

} Pmeasure;
```

Pmeas_process **MEASURE PROCESS**

```
typedef struct {

   Pin_class          dev_class;
   Pint               process_id;

} Pmeas_process;
```

Pmeas_process_set **MEASURE PROCESS SET**

```
typedef struct {

   Pint               set_maxsize;    /* maximum set size            */
   Pmeas_process      process_id;     /* measure process identifier  */

} Pmeas_process_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pmeas_process_list` **MEASURE PROCESS LIST**

```
typedef struct _Pmeas_process_list {

   Pint              count;          /* number of process in list   */
   Pmeas_process     *measure_procs; /* list of measure processes   */

} Pmeas_process_list;
```

`Pmipmap_facs` **MIPMAP FACILITIES**

```
typedef struct {

   Pint      max_dims[3];       /* max dimensions mipmap base level   */
   Pboolean  equal_dims_req;    /* equal mipmap dimensions required   */
   Pboolean  power_2_dims_req; /* power of 2 mipmap dimensions req.  */

} Pmipmap_facs;
```

`Pna_in_status` **NON-ATOMIC INPUT STATUS**

```
typedef enum {

   PNA_IN_STATUS_NONE = 0,
   PNA_IN_STATUS_INCOMPLETE = 1,
   PNA_IN_STATUS_OK = 2

} Pna_in_status;
```

`Pnum_na_in` **NUMBER NON-ATOMIC INPUT DEVICES**

```
typedef struct {

   Pint         set;             /* number available set LIDs         */
   Pint         composite;       /* number available composite LIDs   */

} Pnum_na_in;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Ppattern_mask **PATTERN MASK**

```
typedef struct {
   unsigned int pat_bundle:1;
   unsigned int pat_rep_plus:1;
   unsigned int ext_pat_rep:1;
   unsigned int :29
} Ppattern_mask;
```

Pperspect_corr **PERSPECTIVE CORRECTION**

```
typedef enum {
   PPERSP_CORR_NONE = 0,
   PPERSP_CORR_AT_VERTICES = 1,
   PPERSP_CORR_INTERIOR = 2
} Pperspect_corr;
```

Ppick_mapping_facs **PICK MAPPING FACILITIES**

```
typedef struct {
   Pint_list  avail_pick_types;     /* list available pick types     */
   Pint_list  avail_echo_types;     /* list of available echo types */
} Ppick_mapping_facs;
```

Ppick_path_list **PICK PATH LIST**

```
typedef struct _Ppick_path_list {
   Pint         num_paths;       /* number paths in list            */
   Ppick_path   *paths;          /* array of pick paths             */
} Ppick_path_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Ppick_stat **PICK STATUS**

```
typedef enum {
    PPICK_STAT_FALSE = 0,
    PPICK_STAT_TRUE = 1
} Ppick_stat;
```

Ppict_stat **PICTURE STATUS**

```
typedef enum {
    PPICT_STAT_COMP = 0,
    PPICT_STAT_INCOMP = 1
} Ppict_stat;
```

Ppolyline_mask **POLYLINE MASK**

```
typedef struct {
    unsigned int type:1;
    unsigned int width:1;
    unsigned int colr:1;
    unsigned int shad_method:1;
    unsigned int curve_approx_crit_data:1;
    unsigned int adapt:1;
    unsigned int cont:1;
    unsigned int offset:1;
    unsigned int cap:1;
    unsigned int join:1;
    unsigned int limit:1;
    unsigned int :21;
} Ppolyline_mask;
```

Ppolymarker_mask **POLYMARKER MASK**

```
typedef struct {
    unsigned int type:1;
    unsigned int size:1;
    unsigned int colr:1;
    unsigned int :29;
} Ppolymarker_mask;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pposted_ind **POSTED INDICATOR**

```
typedef enum {

   PPOSTED_FALSE = 0,
   PPOSTED_TRUE = 1

} Pposted_ind;
```

Ppost_grp_facs **POSTING GROUP FACILITIES**

```
typedef struct {

   Pint        num_def_grps;       /* number of definable posting groups
                                      */
   Pint        num_predef_grps;    /* number of predefined posting groups
                                      */
   Pint_list   predef_grps;        /* list of predefined posting groups
                                      */
   Pint_list   backg_methods;      /* list of supported posting group */
                                   /* background methods              */

} Ppost_grp_facs;
```

Ppost_grp_status **POSTING GROUP STATUS**

```
typedef enum {

   PPGSTAT_INACTIVE = 0,
   PPGSTAT_ACTIVE = 1

} Ppost_grp_status;
```

Pref_type **REFERENCE TYPE**

```
typedef enum {

   PREF_EXECUTE = 0,
   PREF_INSTANCE = 1

} Pref_type;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Prefl_mask` **REFLECTANCE MASK**

```
typedef struct {

   unsigned int refl_model:1;
   unsigned int data:1;
   unsigned int :30;

} Prefl_mask;
```

`Prgba` **RED GREEN BLUE WITH ALPHA**

```
typedef struct {

   Pfloat     red;            /* red intensity                    */
   Pfloat     green;          /* green intensity                  */
   Pfloat     blue;           /* blue intensity                   */
   Pfloat     alpha;          /* alpha                            */

} Prgba;
```

`Pset_data` **SET DATA RECORD**

```
typedef struct _Pset_data {

   Pin_class         dev_class;
   Pint              num_in_set;
   _Pin_class_data   *data;

} Pset_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pset_facs **SET FACILITIES**

```
typedef struct {

   Pint      max_measures,              /* max instances                */
   Pint      max_simul_trig,            /* max simul. trigger procs     */
   Pint      max_simul_echo,            /* max simul. echo procs        */
   Pint      max_simul_ack,             /* max simul. ack. procs        */
   Pint_list loc_meas_proc_ids,         /* avail locator meas proc ids */
   Pint_list stroke_meas_proc_ids,      /* avail stroke meas proc ids   */
   Pint_list val_meas_proc_ids,         /* avail valuator meas proc ids*/
   Pint_list choice_meas_proc_ids,      /* avail choice meas proc ids   */
   Pint_list pick_meas_proc_ids,        /* avail pick meas proc ids     */
   Pint_list string_meas_proc_ids,      /* avail string meas proc ids   */
   Pint_list set_meas_proc_ids,         /* avail set meas proc ids      */
   Pint_list composite_meas_proc_ids, /* avail compos meas proc ids   */
   Pint_list trig_proc_ids,             /* avail trigger proc ids       */
   Pint_list loc_echo_proc_ids,         /* avail locator echo proc ids */
   Pint_list stroke_echo_proc_ids,      /* avail stroke echo proc ids   */
   Pint_list val_echo_proc_ids,         /* avail valuator echo proc ids*/
   Pint_list choice_echo_proc_ids,      /* avail choice echo proc ids   */
   Pint_list pick_echo_proc_ids,        /* avail pick echo proc ids     */
   Pint_list string_echo_proc_ids,      /* avail string echo proc ids   */
   Pint_list set_echo_proc_ids,         /* avail set echo proc ids      */
   Pint_list comp_echo_proc_ids,        /* avail compos echo proc ids   */
   Pint_list ack_proc_ids               /* avail ack. proc ids          */

} Pset_facs;
```

Pset_measure **SET MEASURE**

```
typedef struct _Pset_measure {

   Pin_class           dev_class;     /* input class of set           */
   Pint                set_size;      /* size of measure set          */
   _Pin_class_measure *measure_set;   /* array of measures of same    */
                                      /* class                        */

} Pset_measure;
```

Psupport_indication **SUPPORT INDICATION**

```
typedef enum {

   PSI_NOT_SUPPORTED = 0,
   PSI_SUPPORTED = 1

} Psupport_indication;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ptarg_addr` **TARGET ADDRESS**

```
typedef struct {
   Ptarg_type          ref_targ;      /* reference target             */
   Pint                offset;        /* offset from reference target*/
} Ptarg_addr;
```

`Ptarg_data_st` **TARGET DATA STATE**

```
typedef enum {
   PTARG_DATA_ST_ABSENT = 0,
   PTARG_DATA_ST_PRESENT = 1
} Ptarg_data_st;
```

`Ptarg_empty_st` **TARGET EMPTY**

```
typedef enum {
   PTARG_EMPTY_ST_EMPTY = 0,
   PTARG_EMPTY_ST_NOT_EMPTY = 1
} Ptarg_empty_st;
```

`Ptarget_facs` **TARGET FACILITIES**

```
typedef struct {
   Pint           num_avail_targets; /* number of available targets */
} Ptarget_facs;
```

`Ptarg_op_list` **TARGET OPERATION LIST**

```
typedef struct {
   Pint                num_ops;       /* number of target operations */
   Ptarg_op            *targ_ops;     /* list of target operations   */
} Ptarg_op_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ptarg_type` **TARGET TYPE**

```
typedef enum {
    PBASE_TARG = 0,
    PREND_TARG = 1,
    PDISP_TARG = 2
} Ptarg_type;
```

`Ptarg_type_list` **TARGET TYPE LIST**

```
typedef struct {
    Pint            num_targ_types;/* number of target types      */
    Ptarg_type      *targ_types;   /* list of target types        */
} Ptarg_type_list;
```

`Ptest_comp_rel` **TEST COMPARISON RELATIONAL**

```
typedef enum {
    PTRL_EQUAL = 0,
    PTRL_NOT_EQUAL = 1,
    PTRL_GREATER = 2,
    PTRL_LESS = 3,
    PTRL_GREATER_OR_EQUAL = 4,
    PTRL_LESS_OR_EQUAL = 5
} Ptest_comp_rel;
```

`Ptest_comp_logical` **TEST COMPARISON LOGICAL**

```
typedef enum {
    PTLG_BITWISE_AND = 1,
    PTLG_BITWISE_OR = 2,
    PTLG_BITWISE_XOR = 3
} Ptest_comp_logical;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ptext_mask` **TEXT MASK**

```
typedef struct {

   unsigned int font:1;
   unsigned int precision:1;
   unsigned int char_expan:1;
   unsigned int char_space:1;
   unsigned int colr:1;
   unsigned int :27;

} Ptext_mask;
```

`Ptexture_binding` **TEXTURE BINDING**

```
typedef struct {

   Pint     image_res_id;        /* texture image resource identifier */
   Pint     rendering_order;     /* rendering order                   */

} Ptexture_binding;
```

`Ptexture_facs` **TEXTURE FACILITIES**

```
typedef struct {

   Pint_list  coord_srcs;      /* list avail. coordinate sources     */
   Pint_list  compos_methods;  /* list avail. composition methods    */
   Pint_list  min_methods;     /* list avail. minification methods   */
   Pint_list  mag_methods;     /* list avail. magnification methods  */
   Pint_list  bound_cond;      /* list avail. boundary conditions    */
   Pint_list  clamp_methods;   /* list avail. boundary clamp methods */
   Pint_list  render_orders;   /* list avail. rendering orders       */
   Pint       num_pred_inds;   /* number predefined texture indices  */

} Ptexture_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ptexture_map_facs` **TEXTURE MAPPING FACILITIES**

```
typedef struct {
   Pint      max_simul;          /* max number simultaneously          */
                                 /* appliable textures                 */
   Pint_list perspect_corr;      /* list of available perspective      */
                                 /* correction methods                 */
   Pint_list sampling_freqs;     /* list avail. sampling frequencies   */
   Pint_list opt_hints;          /* list of available resource         */
                                 /* optimization hints                 */
   Ptup_util usage;              /* utilization of texture usage       */
                                 /* priorities                         */
   Pint_list image_res_types;    /* list of image resource types       */
                                 /* available for texture mapping      */

} Ptexture_map_facs;
```

`Ptexture_mask` **TEXTURE MASK**

```
typedef struct {
   unsigned int coord_src:1;
   unsigned int coord_src_data:1;
   unsigned int orientation:1;
   unsigned int compos_method:1;
   unsigned int compos_data:1;
   unsigned int min_method:1;
   unsigned int max_method:1;
   unsigned int bound_conds:1;
   unsigned int clamp_method:1
   unsigned int clamp_data:1
   unsigned int depth_sampl_hint:1
   unsigned int freq_wt_hints:1
   unsigned int image_res_id:1
   unsigned int rendering_order:1
   unsigned int :18

} Ptexture_mask;
```

`Ptexture_param` **TEXTURE PARAMETRIZATION**

```
typedef struct {
   Pcoord_src     coord_src;        /* coordinate source and data  */
   Pmatrix3       orientation;      /* orientation matrix          */

} Ptexture_param;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Ptexture_rep **TEXTURE REPRESENTATION**

```
typedef struct {

   Ptexture_param    param;          /* texture parametrization     */
   Ptexture_compos   compos;         /* texture composition         */
   Ptexture_sampling sampling;       /* texture sampling            */
   Ptexture_binding  binding;        /* texture binding             */

} Ptexture_rep;
```

Ptexture_rep **TEXTURE RESOURCE OPTIMIZATION HEURISTICS**

```
typedef struct {

   Pint              opt_hint;          /* optimization hint        */
   Pint_list         usage_priorities;  /* texture usage priorities */

} Ptexture_res_opt_heur;
```

Ptexture_sampling **TEXTURE SAMPLING**

```
typedef struct {

   Pint          min_method;          /* minification method          */
   Pint          mag_method;          /* magnification method         */
   Pint          bound_conds[3];      /* boundary conditions          */
   Pclamp_method clamp;               /* boundary clamp method & data*/
   Pfloat        depth_sampl_hint;    /* depth sampling hint          */
   Pfloat        freq_wt_hints[3];    /* frequency weight hints       */

} Ptexture_sampling;
```

Ptrav_res **TRAVERSAL RESOURCE**

```
typedef struct {

   Pint              id;          /* traversal resource identifier  */
   Pint              type;        /* traversal resource type        */

} Ptrav_res;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Ptrav_res_facs` **TRAVERSAL RESOURCE FACILITIES**

```
typedef struct {

   Ptrav_res_list avail_trav_res;  /* list of avail. traversal resources
                                      */

} Ptrav_res_facs;
```

`Ptrav_res_list` **TRAVERSAL RESOURCE LIST**

```
typedef struct {

   Pint            num_trav_res;   /* number of traversal resources  */
   Ptrav_res       *trav_res;      /* list of traversal resources    */

} Ptrav_res_list;
```

`Ptrav_type` **TRAVERSAL TYPE**

```
typedef enum {

   PTRAV_RENDER = 0,
   PTRAV_PICK = 1

} Ptrav_type;
```

`Ptrig_process` **TRIGGER PROCESS**

```
typedef struct {

   Ptrig_type          type;
   Pint                process_id;

} Ptrig_process;
```

`Ptrig_process_list` **TRIGGER PROCESS LIST**

```
typedef struct {

   Pint                count;
   Ptrig_process       *procs;

} Ptrig_process_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ptrig_type` **TRIGGER TYPE**

```
typedef enum {

   PTRIGGER_BREAK = 0,
   PTRIGGER_SELECT = 1,
   PTRIGGER_EVENT = 2,
   PTRIGGER_DELETE = 3

} Ptrig_type;
```

`Ptup_util` **TEXTURE USAGE PRIORITIES UTILIZATION**

```
typedef enum {

   PTUP_UTILIZED_NO = 0,
   PTUP_UTILIZED_YES = 1

} Ptup_util;
```

`Pval_facs` **VALUATOR FACILITIES**

```
typedef struct {

   Pint      max_simul_trig,   /* max simul. trigger procs           */
   Pint      max_simul_echo,   /* max simul. echo procs              */
   Pint      max_simul_ack,    /* max simul. acknowledgement procs   */
   Pint_list meas_proc_ids,    /* available measure process ids      */
   Pint_list trig_proc_ids,    /* available trigger process ids      */
   Pint_list echo_proc_ids,    /* available echo process ids         */
   Pint_list ack_proc_ids      /* available acknowledgement process ids
                                  */

} Pval_facs;
```

`Pview_attach` **VIEW ATTACHMENT**

```
typedef struct {

   Pint    view_ws_id;              /* workstation id for view     */
   Pint    view_ind;                /* view index                  */
   Pint    view_action_type;        /* view action type            */
   Pint    transfer_type;           /* measure transfer type       */
   Pint    view_upd_method;         /* view update method          */

} Pview_attach;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pview_status **VIEW_STATUS**

```
typedef enum {
    PVIEW_ST_CORRESPOND = 0,
    PVIEW_ST_DIFFERENT = 1
} Pview_status;
```

Pwatch_enable **WATCH ENABLE FLAG**

```
typedef enum {
    PWATCH_OFF = 0,
    PWATCH_ON = 1
} Pwatch_enable;
```

Pws_st_tables_highl **LENGTHS OF WORKSTATION STATE TABLES HIGHLIGHTING**

```
typedef struct {
    Pint   highl_rep;        /* max. # of highlighting table entries  */
} Pws_st_tables_highl;
```

Pws_st_tables_texture **LENGTHS OF WORKSTATION STATE TABLES TEXTURE**

```
typedef struct {
    Pint   texture_rep;         /* max. # of texture table entries    */
} Pws_st_tables_texture;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12 C *Full PHIGS profile* macro definitions

12.1 Function identifiers

The error functions require a unique mapping of the Full PHIGS functions to a set of numbers. The names for these function identifiers are the same as the bound Full PHIGS names except that the sentinel character has been replaced by `Pfn_`.

Function Name Macro	Function Number
`#define Pfn_ws_type_create`	(268)
`#define Pfn_ws_type_set`	(269)
`#define Pfn_ws_type_get`	(270)
`#define Pfn_ws_type_destroy`	(271)
`#define Pfn_redraw_all_structs_from_grp`	(272)
`#define Pfn_redraw_all_structs_on_targ`	(273)
`#define Pfn_redraw_all_structs_from_grp_on_targ`	(274)
`#define Pfn_upd_targ`	(275)
`#define Pfn_define_post_grp`	(276)
`#define Pfn_undefine_post_grp`	(277)
`#define Pfn_set_post_grp_status`	(278)
`#define Pfn_set_post_grp_priority`	(279)
`#define Pfn_set_post_grp_backg_style`	(280)
`#define Pfn_set_post_grp_backg_method`	(281)
`#define Pfn_set_post_grp_border_indic`	(282)
`#define Pfn_set_post_grp_border_ind`	(283)
`#define Pfn_assoc_image_res`	(284)
`#define Pfn_disassoc_image_res`	(285)
`#define Pfn_set_dc_clip_regions3`	(286)
`#define Pfn_set_dc_clip_regions`	(287)
`#define Pfn_set_targ_manip_mode`	(288)
`#define Pfn_set_targ_dispos`	(289)
`#define Pfn_set_disp_targ`	(290)
`#define Pfn_set_rend_targ`	(291)
`#define Pfn_clear_targ`	(292)
`#define Pfn_copy_targ`	(293)
`#define Pfn_create_targ`	(294)
`#define Pfn_destroy_targ`	(295)
`#define Pfn_set_st_visual_rep`	(296)
`#define Pfn_set_targ_st_visual_rep`	(297)
`#define Pfn_assoc_trav_res`	(298)
`#define Pfn_disassoc_trav_res`	(299)
`#define Pfn_manip_trav_res`	(300)
`#define Pfn_reset_all_trav_res`	(301)
`#define Pfn_ret_window_system_colr`	(302)
`#define Pfn_set_appl_int`	(303)
`#define Pfn_set_appl_real`	(304)
`#define Pfn_circle3`	(305)
`#define Pfn_circle`	(306)
`#define Pfn_circular_arc3`	(307)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_circular_arc                    (308)
#define Pfn_ellipse3                        (309)
#define Pfn_ellipse                         (310)
#define Pfn_elliptical_arc3                 (311)
#define Pfn_elliptical_arc                  (312)
#define Pfn_fill_circle3                    (313)
#define Pfn_fill_circle                     (314)
#define Pfn_circular_arc_close3             (315)
#define Pfn_circular_arc_close              (316)
#define Pfn_fill_ellipse3                   (317)
#define Pfn_fill_ellipse                    (318)
#define Pfn_elliptical_arc_close3           (319)
#define Pfn_elliptical_arc_close            (320)
#define Pfn_set_high1_ind                   (321)
#define Pfn_set_linetype_adapt              (322)
#define Pfn_set_linetype_cont               (323)
#define Pfn_set_linetype_offset             (324)
#define Pfn_set_linecap                     (325)
#define Pfn_set_linejoin                    (326)
#define Pfn_set_linemitre_limit             (327)
#define Pfn_set_edgetype_adapt              (328)
#define Pfn_set_edgetype_cont               (329)
#define Pfn_set_edgetype_offset             (330)
#define Pfn_set_edgecap                     (331)
#define Pfn_set_edgejoin                    (332)
#define Pfn_set_edgemitre_limit             (333)
#define Pfn_set_high1_method                (334)
#define Pfn_set_transparency                (335)
#define Pfn_set_back_transparency           (336)
#define Pfn_set_alpha_src_sel               (337)
#define Pfn_set_alpha_data_sel_ind          (338)
#define Pfn_set_active_textures             (339)
#define Pfn_set_back_active_textures        (340)
#define Pfn_set_texture_perspect_corr       (341)
#define Pfn_set_texture_sampling_freq       (342)
#define Pfn_set_texture_res_opt_heur        (343)
#define Pfn_set_cond_flags                  (344)
#define Pfn_set_cond_flags_from_tests       (345)
#define Pfn_set_line_rep_full               (346)
#define Pfn_set_line_rep_mask               (347)
#define Pfn_set_marker_rep_mask             (348)
#define Pfn_set_text_rep_mask               (349)
#define Pfn_set_int_rep_mask                (350)
#define Pfn_set_edge_rep_full               (351)
#define Pfn_set_edge_rep_mask               (352)
#define Pfn_set_ext_pat_rep                 (353)
#define Pfn_set_pat_rep_mask                (354)
#define Pfn_set_texture_rep                 (355)
#define Pfn_set_texture_rep_mask            (356)
#define Pfn_set_texture_param               (357)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_set_texture_composition                (358)
#define Pfn_set_texture_sampling                   (359)
#define Pfn_set_texture_binding                    (360)
#define Pfn_set_highl_rep                          (361)
#define Pfn_set_transparency_mode                  (362)
#define Pfn_set_transparency_thresholds            (363)
#define Pfn_set_refl_rep_mask                      (364)
#define Pfn_set_appl_filter                        (365)
#define Pfn_create_mipmap_texture                  (366)
#define Pfn_define_linetype                        (367)
#define Pfn_define_marker_type                     (368)
#define Pfn_set_view_ref_point3                    (369)
#define Pfn_set_view_ref_point                     (370)
#define Pfn_set_view_plane_norm                    (371)
#define Pfn_set_view_up_vec3                       (372)
#define Pfn_set_view_up_vec                        (373)
#define Pfn_set_view_win_limits                    (374)
#define Pfn_set_proj_vp3                           (375)
#define Pfn_set_proj_vp                            (376)
#define Pfn_set_proj_type                          (377)
#define Pfn_set_proj_ref_point                     (378)
#define Pfn_set_view_plane_dist                    (379)
#define Pfn_set_front_plane_dist                   (380)
#define Pfn_set_back_plane_dist                    (381)
#define Pfn_set_xy_clip_indicator                  (382)
#define Pfn_set_front_clip_indicator               (383)
#define Pfn_set_back_clip_indicator                (384)
#define Pfn_upd_view_rep                           (385)
#define Pfn_map_dc_to_wsc                          (386)
#define Pfn_map_wsc_to_dc                          (387)
#define Pfn_map_dc_to_wc                           (388)
#define Pfn_open_di_struct                         (389)
#define Pfn_close_di_struct                        (390)
#define Pfn_inst_struct                            (391)
#define Pfn_cond_exec_struct                       (392)
#define Pfn_cond_inst_struct                       (393)
#define Pfn_cond_return                            (394)
#define Pfn_cond_skip_elements                     (395)
#define Pfn_cond_skip_to_label                     (396)
#define Pfn_push_st                                (397)
#define Pfn_pop_st                                 (398)
#define Pfn_copy_elem_struct                       (399)
#define Pfn_copy_elem_range_struct                 (400)
#define Pfn_copy_elems_between_labels_struct       (401)
#define Pfn_move_elem                              (402)
#define Pfn_move_elem_range                        (403)
#define Pfn_move_elems_between_labels              (404)
#define Pfn_set_watch_on_elem_range                (405)
#define Pfn_end_watch_on_elem_range                (406)
#define Pfn_post_struct_to_grp                     (407)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define Pfn_unpost_structs_from_grps          (408)
#define Pfn_unpost_all_structs_from_grp       (409)
#define Pfn_post_di_struct                    (410)
#define Pfn_post_di_struct_to_grp             (411)
#define Pfn_unpost_di_struct                  (412)
#define Pfn_renew_di_state                    (413)
#define Pfn_set_di_mode                       (414)
#define Pfn_mark_multi_pass_start             (415)
#define Pfn_mark_multi_pass_compl             (416)
#define Pfn_mark_pass_start                   (417)
#define Pfn_mark_pass_compl                   (418)
#define Pfn_ret_num_passes_req                (419)
#define Pfn_set_set_pick_filter               (420)
#define Pfn_set_composite_pick_filter         (421)
#define Pfn_set_di_pick_filter                (422)
#define Pfn_set_di_pick_corr_point3           (423)
#define Pfn_set_di_pick_corr_point            (424)
#define Pfn_set_pick_mapping_data             (425)
#define Pfn_map_dc_point_to_pick_paths        (426)
#define Pfn_define_locator                    (427)
#define Pfn_define_stroke                     (428)
#define Pfn_define_valuator                   (429)
#define Pfn_define_choice                     (430)
#define Pfn_define_pick                       (431)
#define Pfn_define_string                     (432)
#define Pfn_create_set_measure                (433)
#define Pfn_define_set                        (434)
#define Pfn_create_composite_measure          (435)
#define Pfn_define_composite                  (436)
#define Pfn_undefine_locator                  (437)
#define Pfn_undefine_stroke                   (438)
#define Pfn_undefine_valuator                 (439)
#define Pfn_undefine_choice                   (440)
#define Pfn_undefine_pick                     (441)
#define Pfn_undefine_string                   (442)
#define Pfn_destroy_set_measure               (443)
#define Pfn_undefine_set                      (444)
#define Pfn_destroy_composite_measure         (445)
#define Pfn_undefine_composite                (446)
#define Pfn_init_set3                         (447)
#define Pfn_init_set                          (448)
#define Pfn_init_composite3                   (449)
#define Pfn_init_composite                    (450)
#define Pfn_set_set_mode                      (451)
#define Pfn_set_composite_mode                (452)
#define Pfn_req_set3                          (453)
#define Pfn_req_set                           (454)
#define Pfn_req_composite3                    (455)
#define Pfn_req_composite                     (456)
#define Pfn_sample_set3                       (457)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_sample_set                          (458)
#define Pfn_sample_composite3                   (459)
#define Pfn_sample_composite                    (460)
#define Pfn_get_set3                            (461)
#define Pfn_get_set                             (462)
#define Pfn_get_composite3                      (463)
#define Pfn_get_composite                       (464)
#define Pfn_attach_lid_to_view                  (465)
#define Pfn_detach_lid_from_view                (466)
#define Pfn_attach_lid_to_light_src             (467)
#define Pfn_detach_lid_from_light_src           (468)
#define Pfn_init_di_pick3                       (469)
#define Pfn_init_di_pick                        (470)
#define Pfn_enable_di_pick                      (471)
#define Pfn_disable_di_pick                     (472)
#define Pfn_trans_di_pick_set                   (473)
"
```

Pages 76

12.2 Error codes

```
/* State Errors */
#define PE_NOT_DISO                     (8)   /* Ignoring function, func-
                                                 tion requires state
                                                 (PHOP,*,DISO,*) */
#define PE_NOT_RETAINED_STOP_ONLY       (9)   /* Ignoring function, func-
                                                 tion requires state
                                                 (PHOP,*,STOP,*) */
#define PE_NOT_WSOP_AND_DISO            (10)  /* Ignoring function, func-
                                                 tion requires state
                                                 (PHOP,WSOP,DISO,*) */

/* Workstation Errors */
#define PE_BAD_TARG_OP                  (65)  /* Ignoring function, one of
                                                 the target operations is
                                                 not supported */
#define PE_INVALID_TRAV_RES             (66)  /* Ignoring function, the
                                                 resource is not valid */
#define PE_TRAV_RES_ASSOC_TARG          (67)  /* Ignoring function, the
                                                 traversal resource is
                                                 already associated with a
                                                 target */
#define PE_TRAV_RES_NO_DISASSOC         (68)  /* Ignoring function, the
                                                 resource cannot be disas-
                                                 sociated */
#define PE_INCOMPAT_TRAV_RES            (69)  /* Ignoring function, source
                                                 and destination traversal
                                                 resources are incompatible
                                                 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_DI_STRUCT_NOT_POSTED          (71)  /* Ignoring function, the
                                                  direct interpretation
                                                  structure is not posted on
                                                  the specified workstation
                                                  */
#define PE_DI_REND_TRAV_PROC_ACTIVE      (72)  /* Ignoring function, the
                                                  direct interpretation ren-
                                                  der traversal process is
                                                  already active on the
                                                  specified workstation */
#define PE_DI_REND_TRAV_PROC_NOT_ACTIVE
                                         (73)  /* Ignoring function, the
                                                  direct interpretation tra-
                                                  versal process is not
                                                  active on the specified
                                                  workstation */
#define PE_CANNOT_CREATE_TARGET          (74)  /* Ignoring function, an
                                                  additional target could
                                                  not be created */
#define PE_NUM_TARGETS_NOT_REDUCED       (75)  /* Ignoring function, the
                                                  number of targets cannot
                                                  be reduced below the
                                                  default number available
                                                  for the respective work-
                                                  station type */
#define PE_TARG_NOT_DESTROYED            (76)  /* Ignoring function, the
                                                  target could not be
                                                  destroyed */
#define PE_DISP_REND_TARG_NOT_DESTROYED  (77)  /* Ignoring function, nei-
                                                  ther the display target
                                                  nor the rendering target
                                                  can be destroyed */
#define PE_MULTIPASS_IN_PROGRESS         (78)  /* Ignoring function, a
                                                  multi-pass operation is
                                                  already in progress */
#define PE_MULTI_PASS_NOT_IN_PROGRESS    (79)  /* Ignoring function, a
                                                  multi-pass operation is
                                                  not in progress */
#define PE_PASS_DEF_IN_PROGRESS          (80)  /* Ignoring function, a pass
                                                  definition is already in
                                                  progress */
#define PE_PASS_DEF_NOT_IN_PROGRESS      (81)  /* Ignoring function, a pass
                                                  definition is not in
                                                  progress */
#define PE_LT_SRC_WS_NO_OUT_CAPABILITY   (82)  /* Ignoring function, the
                                                  specified light source
                                                  workstation does not have
                                                  output capability (i.e,
                                                  the workstation category
                                                  is neither OUTPUT, OUTIN,
                                                  nor MO) */
#define PE_POST_GRP_DEF                  (83)  /* Ignoring function, the
                                                  posting group has already
                                                  been defined */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_POST_GRP_NOT_DEF                (84)  /* Ignoring function, the
                                                    posting group has not been
                                                    defined */
#define PE_POST_GRP_ID_LESS_THAN_ONE       (85)  /* Ignoring function, the
                                                    posting group identifier
                                                    is less than one */
#define PE_MAX_POST_GRP_EXCEEDED           (86)  /* Ignoring function, defin-
                                                    ing this posting group
                                                    would exceed the maximum
                                                    number of posting groups
                                                    supported on the worksta-
                                                    tion */
#define PE_IMAGE_RES_NOT_DEF               (87)  /* Ignoring function, image
                                                    resource not defined */
#define PE_BAD_IMAGE_SPEC_METHOD           (88)  /* Ignoring function, the
                                                    image resource image spec-
                                                    ification method is not
                                                    supported for this opera-
                                                    tion */
#define PE_IMAGE_RES_CAP_NOT_SUP           (89)  /* Ignoring function, image
                                                    resource capability is not
                                                    supported on the worksta-
                                                    tion */
#define PE_SPEC_IMAGE_RES_NOT_DEF          (90)  /* Ignoring function, the
                                                    specified image resource
                                                    has not been defined */
#define PE_MAX_IMAGE_RES_EXCEEDED          (91)  /* Ignoring function, associ-
                                                    ating this image resource
                                                    would exceed the maximum
                                                    number of entries allowed
                                                    in the workstation image
                                                    resource table */
#define PE_IMAGE_RES_EXCEEDS_CAP_WS        (92)  /* Ignoring function, the
                                                    image resource exceeds the
                                                    capabilities of the speci-
                                                    fied workstation */
#define PE_VW_WS_OF_CATEGORY_MI            (93)  /* Ignoring function, the
                                                    specified view worksta-
                                                    tion is of category MI */
#define PE_LID_WS_NOT_IN_NOR_OUTIN         (94)  /* Ignoring function, the
                                                    specified LID workstation
                                                    is neither of category
                                                    INPUT nor of category
                                                    OUTIN */
#define PE_NEW_WS_TYPE_NOT_REALIZED        (95)  /* Warning, Not all
                                                    attributes were fully
                                                    realized during the cre-
                                                    ation of the new worksta-
                                                    tion type */
#define PE_WS_TYPE_CANNOT_BE_MODIFIED      (96)  /* Ignoring function, work-
                                                    station type is a default
                                                    type or bound to a work-
                                                    station and cannot be mod-
                                                    ified */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* Output Attribute Errors */
#define PE_NO_STATE_SAVED                (141) /* Ignoring function, no state
                                                  has been saved */
#define PE_SPEC_PAT_TYPE_NOT_SUP         (142) /* Ignoring function, the
                                                  specified pattern type is
                                                  not supported on the spec-
                                                  ified workstation */
#define PE_PAT_TYPE_NOT_SUP_ON_INQ       (143  /* Ignoring function, pattern
                                                  type of requested pattern
                                                  table entry is not sup-
                                                  ported by this inquiry */
#define PE_MAX_VIEWS_EXCEEDED            (150) /* Ignoring function, setting
                                                  this view table entry
                                                  would exceed the maximum
                                                  number of entries allowed
                                                  */
#define PE_INVALID_LIMITS                (151) /* Ignoring function, XMIN ≥
                                                  XMAX, YMIN ≥ YMAX, ZMIN >
                                                  ZMAX, UMIN ≥ UMAX or VMIN ≥
                                                  VMAX */
#define PE_INVALID_VIEWPORT              (152) /* Ignoring function, invalid
                                                  viewport: XMIN ≥ XMAX, YMIN
                                                  ≥ YMAX, ZMIN > ZMAX */
#define PE_INVALID_VIEW_CLIP_LIM         (153) /* Ignoring function, invalid
                                                  view clipping limits: XMIN
                                                  ≥ XMAX, YMIN ≥ YMAX, ZMIN >
                                                  ZMAX */
#define PE_VIEW_CLIP_LIM_NOT_NPC         (154) /* Ignoring function, the view
                                                  clipping limits are not
                                                  within NPC range */
#define PE_PROJ_VP_LIMITS_NOT_NPC        (155) /* Ignoring function, the pro-
                                                  jection viewport limits
                                                  are not within NPC range
                                                  */
#define PE_WS_WIN_LIM_NOT_NPC            (156) /* Ignoring function, the
                                                  workstation window limits
                                                  are not within NPC range
                                                  */
#define PE_WS_VP_NOT_IN_DISPLAY_SPACE    (157) /* Ignoring function, the
                                                  workstation viewport is
                                                  not within display space
                                                  */
#define PE_INVALID_VIEW_PLANES           (158) /* Ignoring function, front
                                                  plane and back plane dis-
                                                  tances same and z-extent
                                                  of projection viewport
                                                  non-zero */
#define PE_VPN_LEN_ZERO                  (159) /* Ignoring function, the view
                                                  plane normal has length
                                                  zero */
#define PE_VUP_LEN_ZERO                  (160) /* Ignoring function, the view
                                                  up vector has lenght zero
                                                  */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_VUP_VPN_PARALLEL                 (161) /* Ignoring function, view up
                                                     and view plane normal vec-
                                                     tors parallel */
#define PE_PRP_BETW_FRONT_BACK              (162) /* Ignoring function, the
                                                     projection reference point
                                                     is between the front and
                                                     back planes */
#define PE_PRP_ON_VIEW_PLANE                (163) /* Ignoring function, the
                                                     projection referece point
                                                     cannot be positioned on
                                                     the view plane */
#define PE_BACK_PLN_BEFORE_FRONT_PLN        (164) /* Ignoring function, the
                                                     back plane is in front of
                                                     the front plane */
#define PE_HIGHL_IND_LT_ONE                 (165) /* Ignoring function, the
                                                     highlighting index is less
                                                     than one */
#define PE_HIGHL_METHOD_NOT_AVAIL           (166) /* Ignoring function, the
                                                     specified highlighting
                                                     method is not available on
                                                     the specified workstation
                                                     */
#define PE_LINETYPE_NOT_DEFINABLE           (167) /* Ignoring function, the
                                                     specified linetype does
                                                     not reference a definable
                                                     linetype */
#define PE_MAX_DEF_LINETYPES_EXCEEDED       (168) /* Ignoring function, set-
                                                     ting this table entry
                                                     would exceed the maximum
                                                     number of entries allowed
                                                     in the definable linetype
                                                     table */
#define PE_LINETYPE_NOT_DEFINED             (169) /* Ignoring function, the
                                                     specified linetype has not
                                                     been defined */
#define PE_LINETYPE_NOT_SUP                 (170) /* Warning, this linetype
                                                     definition is not sup-
                                                     ported on at least one of
                                                     the available workstation
                                                     types */
#define PE_LINETYPE_DEF_NOT_AVAIL           (171) /* Ignoring function, the
                                                     specified standard  or
                                                     implementation-defined
                                                     linetype definition is not
                                                     available */
#define PE_MARKER_TYPE_NOT_DEFINABLE        (172) /* Ignoring function, the
                                                     specified marker type does
                                                     not reference a definable
                                                     marker type */
#define PE_MAX_DEF_MARKER_TYPES_EXCEEDED    (173) /* Ignoring function, set-
                                                     ting this table entry
                                                     would exceed the maximum
                                                     number of entries allowed
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                                    in the definable marker
                                                    type table */
#define PE_MARKER_TYPE_NOT_DEFINED          (174) /* Ignoring function, the
                                                    specified marker type has
                                                    not been defined */
#define PE_MARKER_SHAPE_INVALID             (175) /* Ignoring function, the
                                                    marker shape descriptor is
                                                    invalid */
#define PE_MARKER_SHAPE_TYPE_NOT_SUP        (176) /* Ignoring function, the
                                                    specified marker shape
                                                    type is not supported */
#define PE_MARKER_TYPE_DEF_NOT_AVAIL        (177) /* Ignoring function, the
                                                    specified standard or
                                                    implementation defined
                                                    marker type definition is
                                                    not available */
#define PE_FILTER_SEL_INVALID               (178) /* Ignoring function, the
                                                    filter selection is
                                                    invalid */
#define PE_FILTER_NOT_DEFINED               (179) /* Ignoring function, the
                                                    specified application fil-
                                                    ter has not been defined
                                                    */

/* Input Errors */
#define PE_DI_PICK_TRAV_PROC_ACTIVE         (264) /* Ignoring function, the
                                                    direct interpretation pick
                                                    traversal process is
                                                    already active on the
                                                    specified workstation */
#define PE_DI_PICK_TRAV_PROC_NOT_ACTIVE     (265) /* Ignoring function, the
                                                    direct interpretation pick
                                                    traversal process is not
                                                    active on the specified
                                                    workstation */
#define PE_BAD_PICK_TYPE                    (266) /* Ignoring function, the
                                                    specified pick type is not
                                                    available on the speci-
                                                    fied workstation */
#define PE_LISTED_MEAS_PROC_INVALID         (267) /* Ignoring function, one of
                                                    the listed measure pro-
                                                    cesses is invalid */
#define PE_LISTED_TRIG_PROC_INVALID         (268) /* Ignoring function, one of
                                                    the listed trigger pro-
                                                    cesses is invalid */
#define PE_LISTED_ECHO_PROC_INVALID         (269) /* Ignoring function, one of
                                                    the listed echo processes
                                                    is invalid */
#define PE_LISTED_ACK_PROC_INVALID          (270) /* Ignoring function, one of
                                                    the listed acknowledge-
                                                    ment processes is invalid
                                                    */
#define PE_LID_NR_IN_USE                    (271) /* Ignoring function, device
                                                    number already in use */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_NR_MEAS_INVALID                (272) /* Ignoring function, number
                                                   of measures is invalid */
#define PE_MEAS_ID_NOT_AVAIL              (273) /* Ignoring function, the
                                                   specified measure identi-
                                                   fier is not available on
                                                   the specified workstation
                                                   */
#define PE_MEAS_ID_IN_USE                 (274) /* Ignoring function, the
                                                   specified measure identi-
                                                   fier is currently part of
                                                   a set or composite logi-
                                                   cal input device defini-
                                                   tion */
#define PE_NO_PICK_MEASURE                (275) /* Ignoring function, the
                                                   specified set or compos-
                                                   ite device does not have a
                                                   pick measure */
#define PE_LID_INAPPROP_FOR_VW_ACTION     (276) /* Ignoring function, the
                                                   logical input device class
                                                   specified is inappropri-
                                                   ate for the view action
                                                   specified */
#define PE_LID_INAPPROP_FOR_LT_ACT        (277) /* Ignoring function, the
                                                   input device class speci-
                                                   fied is inappropriate for
                                                   the light action speci-
                                                   fied */
#define PE_LT_ACT_INAPPROP_FOR_LT_SRC     (278) /* Ignoring function, the
                                                   light action is inappro-
                                                   priate for the light
                                                   source specified */

/* Miscellaneous Errors */
#define PE_TEST_OP_INVALID                (451) /* Ignoring function, invalid
                                                   test operator for test type
                                                   */
#define PE_NO_WINDOW_SYSTEM               (452) /* Ignoring function, the
                                                   specified workstation is
                                                   not operating in a window
                                                   management system envrion-
                                                   ment */
#define PE_WIN_SYS_COLR_NOT_DETER         (453) /* Ignoring function, the
                                                   window system colour can-
                                                   not be determined */

/* Additional Output Attribute Errors */
#define PE_TEXTR_COORD_SRC_NOT_AVAIL      (600) /* Ignoring function, the
                                                   specified coordinate
                                                   source is not available on
                                                   the specified workstation
                                                   */
#define PE_DATA_VALUE_IND_LESS_1          (601) /* Ignoring function, the
                                                   specified data value index
                                                   is less than one */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_TEXTR_COMPOS_METHOD_NOT_AVAIL
                                        (602) /* Ignoring function, the
                                                 specified composition
                                                 method is not available on
                                                 the specified workstation
                                                 */
#define PE_TEXTR_MIN_METHOD_NOT_AVAIL   (603) /* Ignoring function, the
                                                 specified minification
                                                 method is not available on
                                                 the specified workstation
                                                 */
#define PE_TEXTR_MAG_METHOD_NOT_AVAIL   (604) /* Ignoring function, the
                                                 specified magnification
                                                 method is not available on
                                                 the specified workstation
                                                 */
#define PE_BOUND_COND_NOT_AVAIL         (605) /* Ignoring function, the
                                                 specified boundary condi-
                                                 tion is not available on
                                                 the specified workstation
                                                 */
#define PE_BOUND_CLAMP_METHOD_NOT_AVAIL
                                        (606) /* Ignoring function, the
                                                 specified boundary clamp
                                                 method is not available on
                                                 the specified workstation
                                                 */
#define PE_DEPTH_SAMPL_BIAS_OUT_OF_RANGE
                                        (607) /* Ignoring function, the
                                                 specified depth sampling
                                                 bias hint is out of range
                                                 */
#define PE_TEXTR_IMAGE_RES_NOT_DEFINED  (608) /* Ignoring function, the
                                                 specified texture image
                                                 resource identifier is not
                                                 defined on the specified
                                                 workstation */
#define PE_RENDERING_ORDER_NOT_AVAIL    (612) /* Ignoring function, the
                                                 specified rendering order
                                                 is not available on the
                                                 specified workstation */
```

12.3 Miscellaneous

12.3.1 Hatch styles

```
#define PHATCH_HORIZ                    (1)
#define PHATCH_VERT                     (2)
#define PHATCH_POS_SLOPE                (3)
#define PHATCH_NEG_SLOPE                (4)
#define PHATCH_HORIZ_VERT_CROSS         (5)
#define PHATCH_POS_NEG_SLOPE_CROSS      (6)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

12.3.2 Prompt and echo types (additional items)

```
#define PPICK_HIGHL_PRIM                    (4)
#define PPICK_HIGHL_PRIM_GROUP              (5)
#define PPICK_HIGHL_STRUCT_NETWORK          (6)
```

12.3.3 Colour types (additional items)

```
#define PCOLR_RGBA                          (5)
#define PCOLR_CIELUVA                       (6)
#define PCOLR_HSVA                          (7)
#define PCOLR_HLSA                          (8)
```

12.3.4 Direct interpretation pick prompt and echo types

```
#define PDI_PICK_HIGHL                      (1)
```

12.3.5 Pick types

```
#define PPICK_TYPE_WS_DEP                   (1)
#define PPICK_TYPE_PICK_FIRST               (2)
#define PPICK_TYPE_PICK_LAST                (3)
#define PPICK_TYPE_PICK_CLOSEST             (4)
```

12.3.6 Target manipulation modes

```
#define PTARG_MANIP_AUTO                    (1)
#define PTARG_MANIP_MANUAL                  (2)
```

12.3.7 Target operation types

```
#define PSET_REND_TARG                      (1)
#define PSET_DISP_TARG                      (2)
#define PCLEAR_TARG                         (3)
#define PCOPY_TARG                          (4)
```

12.3.8 Traversal resource operation types

```
#define PCLEAR_TRAV_RES                     (0)
#define PCOPY_TRAV_RES                      (1)
```

12.3.9 Traversal resource types

```
#define PTRAV_RES_Z_BUF                     (1)
#define PTRAV_RES_PAINT_BUF                 (2)
```

12.3.10 View update method

```
#define PVUM_APPLY_ALL                      (1)
#define PVUM_APPLY_ROTATION                 (2)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12.3.11 View action types

```
#define PVWA_ROT_ABOUT_U_VRC_AXIS                    (1)
#define PVWA_ROT_ABOUT_V_VRC_AXIS                    (2)
#define PVWA_ROT_ABOUT_N_VRC_AXIS                    (3)
#define PVWA_TRANS_U_VRC_AXIS_CLIP                   (4)
#define PVWA_TRANS_U_VRC_AXIS_NOCLIP                 (5)
#define PVWA_TRANS_V_VRC_AXIS_CLIP                   (6)
#define PVWA_TRANS_V_VRC_AXIS_NOCLIP                 (7)
#define PVWA_TRANS_N_VRC_AXIS                        (8)
#define PVWA_SCALE_WIN_UNIFORMLY                     (9)
#define PVWA_SCALE_VP_UNIFORMLY                      (10)
#define PVWA_SCALE_WIN_WIDTH                         (11)
#define PVWA_SCALE_VP_WIDTH                          (12)
#define PVWA_SCALE_WIN_HEIGHT                        (13)
#define PVWA_SCALE_VP_HEIGHT                         (14)
#define PVWA_TRANS_FRONT_CLIP_PLANE                  (15)
#define PVWA_TRANS_BACK_CLIP_PLANE                   (16)
#define PVWA_ROT_AROUND_VRP_ALONG_CYL_LAT            (17)
#define PVWA_TRANS_EYE_FROM_VRP_OUT                  (18)
#define PVWA_TRANS_EYE_FROM_VRP_UP                   (19)
#define PVWA_ROT_EYE_FROM_VRP_UP                     (20)
#define PVWA_CHG_EYE_POS_WRT_VRP                     (21)
#define PVWA_CHG_VRP_POS_ROT_UP_WRT_PRP              (22)
#define PVWA_CHG_VRP_POS_ROT_RL_WRT_PRP              (23)
```

12.3.12 Measure transfer types

```
#define PMTT_MEASURE_CHANGE                  (1)
#define PMTT_EVENT_WITH_DISCARD              (2)
#define PMTT_EVENT_WITH_PROPAGATION          (3)
```

12.3.13 Background methods

```
#define PBACKGM_COLOUR_TABLE_0               (1)
#define PBACKGM_COLOUR_INDEX                 (2)
#define PBACKGM_COLOUR                       (3)
#define PBACKGM_IMAGE_RESOURCE               (4)
```

12.3.14 Image specification methods

```
#define PIMGSM_UNCOMP_COLR_INDEX_ARRAY       (1)
#define PIMGSM_UNCOMP_COLR_ARRAY             (2)
#define PIMGSM_WINDOW_SYS_BITMAP             (3)
#define PIMGSM_VIDEO_SIGNAL_CHAN_ID          (4)
#define PIMGSM_LUMINANCE                     (5)
#define PIMGSM_LUMINANCE_ALPHA               (6)
#define PIMGSM_MIPMAP_COLRS                  (7)
#define PIMGSM_MIPMAP_LUMINANCE              (8)
#define PIMGSM_MIPMAP_LUMINANCE_ALPHA        (9)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

12.3.15 Pattern types

```
#define PPATT_COLOUR_INDEX              (1)
#define PPATT_COLOUR                    (2)
#define PPATT_IMAGE_RESOURCE            (3)
```

12.3.16 Marker Shape Types

```
#define PSHAPE_POLYLINE                 (1)
#define PSHAPE_FILL_AREA                (2)
#define PSHAPE_CONVEX_FILL_AREA         (3)
```

12.3.17 Test Methods

```
#define PTEST_ALWAYS_FAIL               (0)
#define PTEST_ALWAYS_SUCCEED            (1)
#define PTEST_EXTENT3                   (2)
#define PTEST_EXTENT                    (3)
#define PTEST_BOUNDS3                   (4)
#define PTEST_BOUNDS                    (5)
#define PTEST_NAMESET                   (6)
#define PTEST_APPL_INTEGER              (7)
#define PTEST_APPL_REAL                 (8)
```

12.3.18 Filter Types

```
#define PFILTER_TYPE_INVIS              (1)
#define PFILTER_TYPE_HIGHL              (2)
#define PFILTER_TYPE_PICK               (3)
#define PFILTER_TYPE_APPL               (4)
```

12.3.12 Texture sampling frequency types

```
#define PTEXTRSF_PER_PIXEL              (1)
#define PTEXTRSF_INTERP_DEP             (2)
```

12.3.13 Texture optimization hints

```
#define PTEXTR_OPT_NO_HINT              (1)
#define PTEXTR_OPT_SPEED                (2)
#define PTEXTR_OPT_SPACE                (3)
```

12.3.14 Alpha sources

```
#define PALPHASRC_TRANSPARENCY          (1)
#define PALPHASRC_COLR_ASPECT           (2)
#define PALPHASRC_FACET_COLR            (3)
#define PALPHASRC_FACET_DATA            (4)
#define PALPHASRC_VERTEX_COLR           (5)
#define PALPHASRC_VERTEX_DATA           (6)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12.3.15 Texture coordinate sources

```
#define PCOORDSRC_VERTEX_COORD_MC            (1)
#define PCOORDSRC_VERTEX_COORD_WC            (2)
#define PCOORDSRC_VERTEX_NORMAL_MC           (3)
#define PCOORDSRC_VERTEX_NORMAL_WC           (4)
#define PCOORDSRC_NURB_SURF_UV               (5)
#define PCOORDSRC_DATA                       (6)
#define PCOORDSRC_REFL_SPHERE_VRC            (7)
#define PCOORDSRC_REFL_SPHERE_WC             (8)
```

12.3.16 Gamma channel selectors

```
#define PGCS_GAMMA_ONE                       (1)
#define PGCS_GAMMA_RED                       (1)
#define PGCS_GAMMA_TWO                       (2)
#define PGCS_GAMMA_GREEN                     (2)
#define PGCS_GAMMA_THREE                     (3)
#define PGCS_GAMMA_BLUE                      (3)
```

12.3.17 Texture composition methods

```
#define PCOMPOS_REPLACE                      (1)
#define PCOMPOS_MODULATE                     (2)
#define PCOMPOS_BLEND_ENVIRONMENT_COLR       (3)
#define PCOMPOS_DECAL                        (4)
#define PCOMPOS_DECAL_BACKGROUND             (5)
```

12.3.18 Minification methods

```
#define PMINM_SINGLE_BASE                    (1)
#define PMINM_LINEAR_BASE                    (2)
#define PMINM_SINGLE_IN_MIPMAP               (3)
#define PMINM_LINEAR_IN_MIPMAP               (4)
#define PMINM_SINGLE_BETWEEN_MIPMAPS         (5)
#define PMINM_LINEAR_BETWEEN_MIPMAPS         (6)
```

12.3.19 Magnificiation methods

```
#define PMAGM_SINGLE_BASE                    (1)
#define PMAGM_LINEAR_BASE                    (2)
#define PMAGM_SINGLE_IN_MIPMAP               (3)
#define PMAGM_LINEAR_IN_MIPMAP               (4)
#define PMAGM_SINGLE_BETWEEN_MIPMAPS         (5)
#define PMAGM_LINEAR_BETWEEN_MIPMAPS         (6)
```

12.3.20 Boundary conditions

```
#define PBOUND_COND_CLAMP_EXPLICIT_METHOD        (1)
#define PBOUND_COND_CLAMP_BORDER                 (2)
#define PBOUND_COND_WRAP                         (3)
#define PBOUND_COND_MIRROR                       (4)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

12.3.21 Boundary clamp methods

```
#define PBOUND_CLAMP_DISCONTINUE        (1)
#define PBOUND_CLAMP_COLR               (2)
```

12.3.22 Rendering orders

```
#define PRENDORDR_PRELIGHT              (1)
#define PRENDORDR_POST_LIGHT            (2)
```

12.3.23 Transparency modes

```
#define PTRANSPMODE_NONE                (1)
#define PTRANSPMODE_SCREEN_DOOR         (2)
#define PTRANSPMODE_ALPHA_BLEND         (3)
#define PTRANSPMODE_TWO_PASS            (4)
#define PTRANSPMODE_MULTIPASS           (5)
```

12.3.24 Light source action types

```
#define PLTA_ADJ_LT_COLR_COMP_1         (1)
#define PLTA_ADJ_LT_COLR_COMP_2         (2)
#define PLTA_ADJ_LT_COLR_COMP_3         (3)
#define PLTA_ADJ_BRIGHT                 (4)
#define PLTA_ROT_AROUND_X_DIR_VEC       (5)
#define PLTA_ROT_AROUND_Y_DIR_VEC       (6)
#define PLTA_ROT_AROUND_Z_DIR_VEC       (7)
#define PLTA_TRANS_X_LT_POS             (8)
#define PLTA_TRANS_Y_LT_POS             (9)
#define PLTA_TRANS_Z_LT_POS             (10)
#define PLTA_ADJ_CONCENTRATION_EXP      (11)
#define PLTA_ADJ_LT_SPREAD_ANGLE        (12)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13 C *Full PHIGS profile* functions

13.1 Control functions

WORKSTATION TYPE CREATE

```
void pws_type_create (
   Pint                    ws_type,        /* workstation type          */
   const Pconfig_setting   *settings_list,/* array of config. settings*/
   Pint                    *new_ws_type    /* OUT new workstation type */
);
```

WORKSTATION TYPE SET

```
void pws_type_set (
   Pint                    ws_type,        /* workstation type          */
   const Pconfig_setting   *settings_list /* array of config. settings*/
);
```

WORKSTATION TYPE GET

```
void pws_type_get (
   Pint            ws_type,          /* workstation type                */
   const char      *setting_name,    /* configuration setting name      */
   Pstore          store,            /* store object for returned value */
   Pconfig_value   **setting_value   /* OUT configuration setting value */
);
```

WORKSTATION TYPE DESTROY

```
void pws_type_destroy (
   Pint         ws_type                    /* workstation type          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

REDRAW ALL STRUCTURES FROM POSTING GROUP

```
void predraw_all_structs_from_grp (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* posting group               */
   Pbackg_redisplay  redisplay_flag  /* background redisplay flag   */

);
```

REDRAW ALL STRUCTURES ON TARGET

```
void predraw_all_structs_on_targ (

   Pint              ws_id,          /* workstation identifier      */
   const Ptarg_addr  *targ_addr,     /* target address              */
   Pctrl_flag        ctrl_flag       /* control flag                */

);
```

REDRAW ALL STRUCTURES FROM POSTING GROUP ON TARGET

```
void predraw_all_from_grp_on_targ (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* posting group               */
   Pbackg_redisplay  redisplay_flag  /* background redisplay flag   */
   const Ptarg_addr  *targ_addr,     /* target address              */
   Pctrl_flag        ctrl_flag       /* control flag                */

);
```

UPDATE TARGET

```
void pupd_targ (

   Pint              ws_id,          /* workstation identifier      */
   const Ptarg_addr  *targ_addr,     /* target address              */
   Pregen_flag       regen_flag      /* update regeneration flag    */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

DEFINE POSTING GROUP

```
void pdefine_post_grp (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id,          /* posting group identifier    */
   Pint                ref_grp_id,      /* reference posting group id  */
   Prel_pri            rel_pri,         /* relative priority           */
   Pint                def_view_ind,    /* default view index          */
   Ppost_grp_status    post_grp_status, /* posting group status        */
   Pbackg_style        backg_style,     /* posting group backg. style  */
   const Pbackg_method*backg_method,    /* posting group backg. method */
   Pborder_indic       border_indic,    /* posting group border indic. */
   Pint                border_ind       /* posting group border index  */

);
```

UNDEFINE POSTING GROUP

```
void pundefine_post_grp (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id           /* posting group identifier    */

);
```

SET POSTING GROUP STATUS

```
void pset_post_grp_status (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id,          /* posting group identifier    */
   Ppost_grp_status    post_grp_status  /* posting group status        */

);
```

SET POSTING GROUP PRIORITY

```
void pset_post_grp_priority (

   Pint       ws_id,                    /* workstation identifier      */
   Pint       grp_id,                   /* posting group identifier    */
   Pint       ref_grp_id,               /* reference posting group id  */
   Prel_pri   rel_pri                   /* relative priority           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET POSTING GROUP BACKGROUND STYLE

```
void pset_post_grp_backg_style (

   Pint            ws_id,          /* workstation identifier          */
   Pint            grp_id,         /* posting group identifier        */
   Pbackg_style    backg_style     /* posting group background style */

);
```

SET POSTING GROUP BACKGROUND METHOD

```
void pset_post_grp_backg_method (

   Pint                  ws_id,          /* workstation identifier      */
   Pint                  grp_id,         /* posting group identifier    */
   const Pbackg_method   *backg_method   /* posting group background    */
                                         /* method                      */

);
```

SET POSTING GROUP BORDER INDICATOR

```
void pset_post_grp_border_indicator (

   Pint            ws_id,            /* workstation identifier          */
   Pint            grp_id,           /* posting group identifier        */
   Pborder_indic   border_indic      /* posting group border indicator */

);
```

SET POSTING GROUP BORDER INDEX

```
void pset_post_grp_border_ind (

   Pint       ws_id,              /* workstation identifier           */
   Pint       grp_id,             /* posting group identifier         */
   Pint       border_ind          /* posting group border index       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ASSOCIATE IMAGE RESOURCE

```
void passoc_image_res (

   Pint                ws_id,          /* workstation identifier      */
   Pint                image_res_id,   /* image resource identifier   */
   const Pint_size     *dims,          /* m,n dimensions              */
   const Pimage_res    *image_res,     /* image resource              */
   Paccess_flag        *access_flag    /* OUT access flag             */

);
```

DISASSOCIATE IMAGE RESOURCE

```
void pdisassoc_image_res (

   Pint                ws_id,          /* workstation identifier      */
   Pint                image_res_id    /* image resource identifier   */

);
```

SET DEVICE COORDINATE CLIP REGIONS 3

```
void pset_dc_clip_regions3 (

   Pint                  ws_id,        /* workstation identifier      */
   const Plimit3_list    *limit_list   /* list of DC clip regions     */

);
```

SET DEVICE COORDINATE CLIP REGIONS

```
void pset_dc_clip_regions (

   Pint                ws_id,          /* workstation identifier      */
   const Plimit_list   *limit_list     /* list of DC clip regions     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET TARGET MANIPULATION MODE

```
void pset_targ_manip_mode (

   Pint              ws_id,              /* workstation identifier      */
   Pint              targ_manip_mode     /* target manipulation mode    */

);
```

SET TARGET DISPOSITION

```
void pset_targ_dispos (

  Pint                   ws_id,          /* workstation identifier      */
  const Ptarg_op_list    *targ_ops       /* target operations list      */

);
```

SET DISPLAY TARGET

```
void pset_disp_targ (

   Pint                ws_id,           /* workstation identifier      */
   const Ptarg_addr    *targ_addr       /* target address              */

);
```

SET RENDERING TARGET

```
void pset_rend_targ (

   Pint                ws_id,           /* workstation identifier      */
   const Ptarg_addr    *targ_addr       /* target address              */

);
```

CLEAR TARGET

```
void pclear_targ (

   Pint                ws_id,           /* workstation identifier      */
   const Ptarg_addr    *targ_addr       /* target address              */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

COPY TARGET

```
void pcopy_targ (

   Pint              ws_id,           /* workstation identifier      */
   const Ptarg_addr  *src_targ_addr,  /* source target address       */
   const Ptarg_addr  *dest_targ_addr  /* destination target address  */

);
```

CREATE TARGET

```
void pcreate_targ (

   Pint              ws_id,           /* workstation identifier      */
   const Ptarg_addr  *targ_addr       /* target address              */

);
```

DESTROY TARGET

```
void pdestroy_targ (

   Pint              ws_id,           /* workstation identifier      */
   const Ptarg_addr  *targ_addr       /* target address              */

);
```

SET STATE OF VISUAL REPRESENTATION

```
void pset_st_visual_rep (

   Pint              ws_id,          /* workstation identifier          */
   Pvisual_st        visual_st       /* state of visual representation  */

);
```

SET TARGET STATE OF VISUAL REPRESENTATION

```
void pset_targ_st_visual_rep (

   Pint              ws_id,          /* workstation identifier          */
   const Ptarg_addr  *targ_addr,     /* target address                  */
   Pvisual_st        visual_st       /* state of visual representation */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ASSOCIATE TRAVERSAL RESOURCE

```
void passoc_trav_res (

   Pint                ws_id,        /* workstation identifier         */
   Pint                res_id,       /* traversal resource identifier  */
   const Ptarg_addr    *targ_addr    /* source target address          */

);
```

DISASSOCIATE TRAVERSAL RESOURCE

```
void pdisassoc_trav_res (

   Pint                ws_id,        /* workstation identifier         */
   Pint                res_id        /* traversal resource identifier  */

);
```

MANIPULATE TRAVERSAL RESOURCE

```
void pmanip_trav_res (

  Pint                  ws_id,        /* workstation identifier        */
  Pint                  res_id,       /* traversal resource identifier*/
  const Ptrav_res_op    *manip_data   /* traversal resource operation */

);
```

RESET ALL TRAVERSAL RESOURCES

```
void preset_all_trav_res (

  Pint                  ws_id         /* workstation identifier       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

RETRIEVE WINDOW SYSTEM COLOUR

```
void pret_window_system_colr (

  Pint           ws_id,           /* workstation identifier           */
  const Pgcolr   *colr,           /* colour sought                    */
  Pstore         store,           /* store object for returned value  */
  Pdata          **winsys_colr    /* OUT window system colour         */

);
```

SET APPLICATION INTEGER

```
void pset_appl_int (

   Pint     integer_id,     /* application integer identifier       */
   Pint     value           /* value of application integer         */

);
```

SET APPLICATION REAL

```
void pset_appl_real (

   Pint     real_id,        /* application real identifier          */
   Pfloat   value           /* value of application real            */

);
```

13.2 Output primitive functions

CIRCLE 3

```
void pcircle3 (

   const Ppoint3      *ctr_point,      /* centre point                */
   Pfloat             radius,          /* radius                      */
   const Pvec3        ref_vecs[2]      /* reference vectors           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

CIRCLE

```
void pcircle (
   const Ppoint        *ctr_point,      /* centre point                */
   Pfloat              radius           /* radius                      */
);
```

CIRCULAR ARC 3

```
void pcircular_arc3 (
   const Ppoint3       *ctr_point,      /* centre point                */
   Pfloat              radius,          /* radius                      */
   const Pvec3         ref_vecs[2],     /* reference vectors           */
   Pfloat              start,           /* start angle                 */
   Pfloat              end              /* end angle                   */
);
```

CIRCULAR ARC

```
void pcircular_arc (
   const Ppoint        *ctr_point,      /* centre point                */
   Pfloat              radius,          /* radius                      */
   Pfloat              start,           /* start angle                 */
   Pfloat              end              /* end angle                   */
);
```

ELLIPSE 3

```
void pellipse3 (
   const Ppoint3    *ctr_point,         /* centre point                */
   const Pvec3      *major_ref_vec,     /* major axis reference vector */
   const Pvec3      *minor_ref_vec      /* minor axis reference vector */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ELLIPSE

```
void pellipse (
   const Ppoint     *ctr_point,        /* centre point                */
   const Pvec       *major_ref_vec,    /* major axis reference vector */
   const Pvec       *minor_ref_vec     /* minor axis reference vector */
);
```

ELLIPTICAL ARC 3

```
void pelliptical_arc3 (
   const Ppoint3    *ctr_point,        /* centre point                */
   const Pvec3      *major_ref_vec,    /* major axis reference vector */
   const Pvec3      *minor_ref_vec,    /* minor axis reference vector */
   Pfloat           start,             /* start angle                 */
   Pfloat           end                /* end angle                   */
);
```

ELLIPTICAL ARC

```
void pelliptical_arc (
   const Ppoint     *ctr_point,        /* centre point                */
   const Pvec       *major_ref_vec,    /* major axis reference vector */
   const Pvec       *minor_ref_vec,    /* minor axis reference vector */
   Pfloat           start,             /* start angle                 */
   Pfloat           end                /* end angle                   */
);
```

FILL CIRCLE 3

```
void pfill_circle3 (
   const Ppoint3       *ctr_point,     /* centre point                */
   Pfloat              radius,         /* radius                      */
   const Pvec3         ref_vecs[2]     /* reference vectors           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FILL CIRCLE

```
void pfill_circle (
   const Ppoint        *ctr_point,      /* centre point                */
   Pfloat              radius           /* radius                      */
);
```

CIRCULAR ARC CLOSE 3

```
void pcircular_arc_close3 (
   const Ppoint3       *ctr_point,      /* centre point                */
   Pfloat              radius,          /* radius                      */
   const Pvec3         ref_vecs[2],     /* reference vectors           */
   Pfloat              start,           /* start angle                 */
   Pfloat              end,             /* end angle                   */
   Pclosure            type             /* closure type                */
);
```

CIRCULAR ARC CLOSE

```
void pcircular_arc_close (
   const Ppoint        *ctr_point,      /* centre point                */
   Pfloat              radius,          /* radius                      */
   Pfloat              start,           /* start angle                 */
   Pfloat              end,             /* end angle                   */
   Pclosure            type             /* closure type                */
);
```

FILL ELLIPSE 3

```
void pfill_ellipse3 (
   const Ppoint3    *ctr_point,         /* centre point                */
   const Pvec3      *major_ref_vec,     /* major axis reference vector */
   const Pvec3      *minor_ref_vec      /* minor axis reference vector */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FILL ELLIPSE

```
void pfill_ellipse (

   const Ppoint     *ctr_point,        /* centre point                */
   const Pvec       *major_ref_vec,    /* major axis reference vector */
   const Pvec       *minor_ref_vec     /* minor axis reference vector */

);
```

ELLIPTICAL ARC CLOSE 3

```
void pelliptical_arc_close3 (

   const Ppoint3    *ctr_point,        /* centre point                */
   const Pvec3      *major_ref_vec,    /* major axis reference vector */
   const Pvec3      *minor_ref_vec,    /* minor axis reference vector */
   Pfloat           start,             /* start angle                 */
   Pfloat           end,               /* end angle                   */
   Pclosure         type               /* closure type                */

);
```

ELLIPTICAL ARC CLOSE

```
void pelliptical_arc_close (

   const Ppoint     *ctr_point,        /* centre point                */
   const Pvec       *major_ref_vec,    /* major axis reference vector */
   const Pvec       *minor_ref_vec,    /* minor axis reference vector */
   Pfloat           start,             /* start angle                 */
   Pfloat           end,               /* end angle                   */
   Pclosure         type               /* closure type                */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.3 Attribute specification functions

13.3.1 Bundled attribute selection

SET HIGHLIGHTING INDEX

```
void pset_highl_ind (
   Pint        highl_ind                /* highlighting index          */
);
```

13.3.2 Individually selected bundled attributes

SET LINETYPE ADAPTABILITY

```
void pset_linetype_adapt (
   Plinetype_adapt     adaptability     /* linetype adaptability       */
);
```

SET LINETYPE CONTINUITY

```
void pset_linetype_cont (
   Plinetype_cont      continuity       /* linetype continuity         */
);
```

SET LINETYPE OFFSET

```
void pset_linetype_offset (
   Pfloat              offset           /* linetype offset             */
);
```

SET LINECAP

```
void pset_linecap (
   Plinecap            linecap          /* linecap                     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET LINEJOIN

```
void pset_linejoin (
   Plinejoin          linejoin        /* linejoin                    */
);
```

SET LINEMITRE LIMIT

```
void pset_linemitre_limit (
   Pfloat             limit           /* linemitre limit             */
);
```

SET EDGETYPE ADAPTABILITY

```
void pset_edgetype_adapt (
   Pedgetype_adapt    adaptability    /* edgetype adaptability       */
);
```

SET EDGETYPE CONTINUITY

```
void pset_edgetype_cont (
   Pedgetype_cont     continuity      /* edgetype continuity         */
);
```

SET EDGETYPE OFFSET

```
void pset_edgetype_offset (
   Pfloat             offset          /* edgetype offset             */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET EDGECAP

```
void pset_edgecap (
   Pedgecap          edgecap          /* edgecap*/
);
```

SET EDGEJOIN

```
void pset_edgejoin (
   Pedgejoin         edgejoin         /* edgejoin                    */
);
```

SET EDGEMITRE LIMIT

```
void pset_edgemitre_limit (
   Pfloat            limit            /* edgemitre limit             */
);
```

SET HIGHLIGHTING METHOD

```
void pset_highl_method (
   const Phighl_method *method        /* highlighting method         */
);
```

SET TRANSPARENCY

```
void pset_transparency (
   Pfloat    transparency             /* transparency                */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET BACK TRANSPARENCY

```
void pset_back_transparency (
   Pfloat    back_transparency          /* back transparency           */
);
```

SET ALPHA SOURCE SELECTOR

```
void pset_alpha_src_sel (
   const Pint_list  *alpha_src_sel      /* alpha source selector list  */
);
```

SET ALPHA DATA SELECTION INDEX

```
void pset_alpha_data_sel_ind (
   Pint        alpha_data_sel_ind       /* alpha data selection index  */
);
```

SET ACTIVE TEXTURES

```
void pset_active_textures (
   const Pint_list  *active_textures    /* list of active textures     */
);
```

SET BACK ACTIVE TEXTURES

```
void pset_back_active_textures (
   const Pint_list  *back_active_textures /* list of active textures  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET TEXTURE PERSPECTIVE CORRECTION

```
void pset_texture_perspect_corr (
   Pperspect_corr   perspect_corr    /* texture perspective correction */
);
```

SET TEXTURE SAMPLING FREQUENCY

```
void pset_texture_sampling_freq (
   Pint          sampling_freq          /* texture sampling frequency  */
);
```

SET TEXTURE RESOURCE OPTIMIZATION HEURISTICS

```
void pset_texture_res_opt_heur (
   const Ptexture_res_opt_heur *heuristics   /* heuristics              */
);
```

13.3.3 Individual attributes

SET CONDITION FLAGS

```
void pset_cond_flags (
   const Pcond_flags     *flags        /* condition flag masks        */
);
```

SET CONDITION FLAGS FROM TESTS

```
void pset_cond_flags_from_tests (
   const Pcond_test_list  *tests       /* list of condition flag tests*/
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

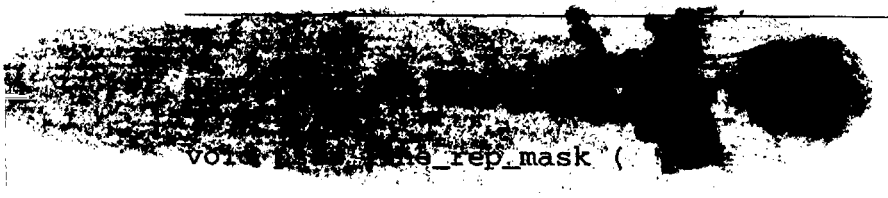
13.3.4 Workstation attribute table definitions

SET POLYLINE REPRESENTATION FULL

```
void pset_line_rep_full (

   Pint                      ws_id,     /* workstation identifier      */
   Pint                      index,     /* polyline bundle index       */
   const Pline_bundle_full   *rep       /* polyline representation full*/

);
```



```
void pset_line_rep_mask (

   Pint                      ws_id,     /* workstation identifier      */
   Pint                      index,     /* polyline bundle index       */
   Ppolyline_mask            mask,      /* polyline validity mask      */
   const Pline_bundle_full   rep        /* polyline representation     */

);
```

SET POLYMARKER REPRESENTATION MASK

```
void pset_marker_rep_mask (

   Pint                      ws_id,     /* workstation identifier      */
   Pint                      index,     /* polymarker bundle index     */
   Ppolymarker_mask          mask,      /* polymarker validity mask    */
   const Pmarker_bundle_plus rep        /* polymarker representation   */

);
```

SET TEXT REPRESENTATION MASK

```
void pset_text_rep_mask (

   Pint                      ws_id,     /* workstation identifier      */
   Pint                      index,     /* text bundle index           */
   Ptext_mask                mask,      /* text validity mask          */
   const Ptext_bundle_plus   rep        /* text representation         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET INTERIOR REPRESENTATION MASK

```
void pset_int_rep_mask (

   Pint                        ws_id,    /* workstation identifier      */
   Pint                        index,    /* interior bundle index       */
   Pinterior_mask              mask,     /* interior validity mask      */
   const Pint_bundle_plus      rep       /* interior representation     */

);
```

SET EDGE REPRESENTATION FULL

```
void pset_edge_rep_full (

   Pint                        ws_id,    /* wor                         */
   Pint                        index,    /* edge bundle index           */
   const Pedge_bundle_full     *rep      /* edge representation full    */

);
```

SET EDGE REPRESENTATION MASK

```
void pset_edge_rep_mask (

   Pint                        ws_id,    /* workstation identifier      */
   Pint                        index,    /* edge bundle index           */
   Pedge_mask                  mask,     /* edge validity mask          */
   const Pedge_bundle_full     *rep      /* edge representation full    */

);
```

SET EXTENDED PATTERN REPRESENTATION

```
void pset_ext_pat_rep (

   Pint              ws_id,       /* workstation identifier          */
   Pint              pat_ind,     /* pattern bundle index            */
   const Ppattern    *pattern     /* extended pattern representation */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET PATTERN REPRESENTATION MASK

```
void pset_pat_rep_mask (

   Pint                ws_id,          /* workstation identifier      */
   Pint                pat_ind,        /* pattern bundle index        */
   Ppattern_mask       mask,           /* pattern validity mask       */
   const Ppat_rep      *pat_bundle,    /* pattern representation      */
   const Ppat_rep_plus *pat_rep_plus,  /* pattern representation plus */
   const Ppattern      *pattern        /* ext. pattern representation */

);
```

NOTE — Only the parameters indicated by the mask need be provided. Other representation parameters may reference null structures or unused representation structures;

SET TEXTURE REPRESENTATION

```
void pset_texture_rep (

   Pint                    ws_id,     /* workstation identifier      */
   Pint                    index,     /* texture index               */
   const Ptexture_rep      *rep       /* texture representation      */

);
```

SET TEXTURE REPRESENTATION MASK

```
void pset_texture_rep_mask (

   Pint                    ws_id,     /* workstation identifier      */
   Pint                    index,     /* texture index               */
   Ptexture_mask           mask,      /* texture validity mask       */
   const Ptexture_rep      *rep       /* texture representation      */

);
```

NOTE — Only those fields of the structures within the texture representation that are specified by the mask are referenced.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET TEXTURE PARAMETRIZATION

```
void pset_texture_param (

   Pint                    ws_id,   /* workstation identifier          */
   Pint                    index,   /* texture index                   */
   const Ptexture_param    *param   /* texture parametrization         */

);
```

SET TEXTURE COMPOSITION

```
void pset_texture_composition (

   Pint                     ws_id,      /* workstation identifier      */
   Pint                     index,      /* texture index               */
   const Ptexture_compos    *compos     /* texture composition         */

);
```

SET TEXTURE SAMPLING

```
void pset_texture_sampling (

   Pint                      ws_id,     /* workstation identifier      */
   Pint                      index,     /* texture index               */
   const Ptexture_sampling   *sampling  /* texture sampling            */

);
```

SET TEXTURE BINDING

```
void pset_texture_binding (

   Pint                     ws_id,      /* workstation identifier      */
   Pint                     index,      /* texture index               */
   const Ptexture_binding   *binding    /* texture binding             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET HIGHLIGHTING REPRESENTATION

```
void pset_highl_rep (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 highl_ind,      /* highlighting bundle index   */
   const Phighl_method  *method         /* highlighting method         */

);
```

SET TRANSPARENCY MODE

```
void pset_transparency_mode (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    mode         /* transparency mode           */

);

);
```

SET DEPTH CUE REPRESENTATION MASK

```
void pset_depth_cue_rep_mask (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    index,       /* depth cue bundle index      */
   Pdepth_cue_mask         mask,        /* depth cue validity mask     */
   const Pdepth_cue_rep    *rep         /* depth cue representation    */

);
```

SET TRANSPARENCY THRESHOLDS

```
void pset_transparency_thresholds (

   Pint                ws_id,       /* workstation identifier          */
   const Pfloat_list   *thresholds  /* list of transparency thresholds */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET REFLECTANCE REPRESENTATION MASK

```
void pset_refl_rep_mask (

   Pint                    ws_id,      /* workstation identifier      */
   Pint                    index,      /* reflectance bundle index    */
   Prefl_mask              mask,       /* reflectance validity mask   */
   const Prefl_rep         *rep        /* reflectance representation  */

);
```

13.3.5 Workstation attribute filter definition

SET APPLICATION FILTER

```
void pset_appl_filter (

   Pint           ws_id,           /* workstation identifier         */
   Pint           filter_id,       /* application filter identifier  */
   const Pfilter  *filter          /* application filter              */

);
```

13.3.6 Attribute utility functions

CREATE MIPMAP TEXTURE

```
void pcreate_mipmap_texture (

   Pint     ws_id,                 /* workstation identifier          */
   Pint     src_image_res_id,      /* source image resource id        */
   Pint     mipmap_image_res_id,   /* mipmap image resource id        */
   Pint     max_levels,            /* max mipmap levels               */
   Pint     creation_heuristic     /* mipmap creation heuristic       */

);
```

DEFINE LINETYPE

```
void pdefine_linetype (

   Pint                linetype,          /* linetype                 */
   Pfloat              repeat_length,     /* repeat length            */
   const Pfloat_list   *segment_lengths   /* segment length list      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

DEFINE MARKER TYPE

```
void pdefine_marker_type (
   Pint                marker_type,     /* marker type                */
   const Pmarker_desc  *marker          /* marker descriptor          */
);
```

13.4 Transformation and clipping functions

13.4.1 View construction functions

SET VIEW REFERENCE POINT 3

```
void pset_view_ref_point3 (
   Pint              ws_id,             /* workstation id             */
   Pint              view_ind,          /* view index                 */
   const Ppoint3     *view_ref_point    /* view reference point       */
);
```

SET VIEW REFERENCE POINT

```
void pset_view_ref_point (
   Pint              ws_id,             /* workstation id             */
   Pint              view_ind,          /* view index                 */
   const Ppoint      *view_ref_point    /* view reference point       */
);
```

SET VIEW PLANE NORMAL

```
void pset_view_plane_norm (
   Pint              ws_id,             /* workstation id             */
   Pint              view_ind,          /* view index                 */
   const Pvec3       *view_norm_vec     /* view normal vector         */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET VIEW UP VECTOR 3

```
void pset_view_up_vec3 (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Pvec3       *view_up_vec    /* view up vector              */
);
```

SET VIEW UP VECTOR

```
void pset_view_up_vec (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Pvec        *view_up_vec    /* view up vector              */
);
```

SET VIEW WINDOW LIMITS

```
void pset_view_win_limits (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Plimit      *win            /* view window limits          */
);
```

SET PROJECTION VIEWPORT 3

```
void pset_proj_vp3 (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Plimit3     *proj_vp        /* projection viewport limits  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET PROJECTION VIEWPORT

```
void pset_proj_vp (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   const Plimit      *proj_vp        /* projection viewport limits    */

);
```

SET PROJECTION TYPE

```
void pset_proj_type (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   Pproj_type        proj_type       /* projection type               */

);
```

SET PROJECTION REFERENCE POINT

```
void pset_proj_ref_point (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   const Ppoint3     *proj_ref_point /* projection reference point    */

);
```

SET VIEW PLANE DISTANCE

```
void pset_view_plane_dist (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   Pfloat            view_plane      /* view plane distance           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET FRONT PLANE DISTANCE

```
void pset_front_plane_dist (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pfloat            front_plane     /* front plane distance        */

);
```

SET BACK PLANE DISTANCE

```
void pset_back_plane_dist (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pfloat            back_plane      /* back plane distance         */

);
```

SET X-Y CLIPPING INDICATOR

```
void pset_xy_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pclip_ind         xy_clip         /* xy clipping indicator       */

);
```

SET FRONT CLIPPING INDICATOR

```
void pset_front_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pclip_ind         front_clip      /* front clipping indicator    */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET BACK CLIPPING INDICATOR

```
void pset_back_clip_indicator (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pclip_ind         back_clip       /* back clipping indicator     */
);
```

UPDATE VIEW REPRESENTATION

```
void pupd_view_rep (
   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              view_upd_method /* view update method          */
);
```

13.4.2 Workstation utility functions

MAP DEVICE COORDINATES TO WINDOW SYSTEM COORDINATES

```
void pmap_dc_to_wsc (
   Pint              ws_id,   /* workstation id                     */
   const Ppoint_list *dc,     /* device coordinate points           */
   Ppoint_list       *wsc     /* OUT window system coordinate points */
);
```

MAP WINDOW SYSTEM COORDINATES TO DEVICE COORDINATES

```
void pmap_wsc_to_dc (
   Pint              ws_id,       /* workstation id                 */
   const Ppoint_list *wsc,        /* window system coordinate points */
   Ppoint_list       *dc          /* OUT device coordinate points   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

MAP DEVICE COORDINATES TO WORLD COORDINATES

```
void pmap_dc_to_wc (
   Pint              ws_id,       /* workstation id                    */
   Pint              view_ind,    /* view index                        */
   const Ppoint_list *dc,         /* device coordinate points          */
   Ppoint_list       *wc          /* OUT world coordinate points       */
);
```

13.5 Structure content functions

OPEN DIRECT INTERPRETATION STRUCTURE

```
void popen_di_struct (
   void
);
```

CLOSE DIRECT INTERPRETATION STRUCTURE

```
void pclose_di_struct (
   void
);
```

INSTANCE STRUCTURE

```
void pinst_struct (
   Pint        struct_id              /* structure identifier        */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

CONDITIONAL EXECUTE STRUCTURE

```
void pcond_exec_struct (

   Pint                struct_id,       /* structure identifier        */
   const Ptest         *test            /* condition test              */

);
```

CONDITIONAL INSTANCE STRUCTURE

```
void pcond_inst_struct (

   Pint                struct_id,       /* structure identifier        */
   const Ptest         *test            /* condition test              */

);
```

CONDITIONAL RETURN

```
void pcond_return (

   const Ptest         *test            /* condition test              */

);
```

CONDITIONAL SKIP ELEMENTS

```
void pcond_skip_elements (

   Pint          skip_count,            /* number of elements to skip  */
   const Ptest   *test                  /* condition test              */

);
```

CONDITIONAL SKIP TO LABEL

```
void pcond_skip_to_label (

   Pint          label,                 /* label to which to skip      */
   const Ptest   *test                  /* condition test              */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

PUSH STATE

```
void ppush_st (
   Pint              struct_id,      /* structure identifier        */
   Pref_type         ref_type        /* reference type              */
);
```

POP STATE

```
void ppop_st (
   void
);
```

COPY ELEMENT FROM STRUCTURE

```
void pcopy_elem_struct (
   Pint          struct_id,          /* structure identifier        */
   Pint          elem_num            /* element number              */
);
```

COPY ELEMENT RANGE FROM STRUCTURE

```
void pcopy_elem_range_struct (
   Pint          struct_id,          /* structure identifier        */
   Pint          elem_ptr_1_value,   /* element position 1          */
   Pint          elem_ptr_2_value    /* element position 2          */
);
```

COPY ELEMENTS BETWEEN LABELS FROM STRUCTURE

```
void pcopy_elems_between_labels_struct (
   Pint          struct_id,          /* structure identifier        */
   Pint          label1_id,          /* label identifier 1          */
   Pint          label2_id           /* label identifier 2          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

MOVE ELEMENT FROM STRUCTURE

```
void pmove_elem_struct (

   Pint          struct_id,          /* structure identifier        */
   Pint          elem_num            /* element number              */

);
```

MOVE ELEMENT RANGE FROM STRUCTURE

```
void pmove_elem_range_struct (

   Pint          struct_id,          /* structure identifier        */
   Pint          elem_ptr_1_value,   /* element position 1          */
   Pint          elem_ptr_2_value    /* element position 2*/

);
```

MOVE ELEMENTS BETWEEN LABELS FROM STRUCTURE

```
void pmove_elems_between_labels_struct (

   Pint          struct_id,          /* structure identifier        */
   Pint          label1_id,          /* label identifier 1          */
   Pint          label2_id           /* label identifier 2          */

);
```

SET WATCH ON ELEMENT RANGE

```
void pset_watch_on_elem_range (

   Pint      struct_id,              /* watched structure identifier*/
   Pint      elem_pos1,              /* element position 1          */
   Pint      elem_pos2               /* element position 2          */

);
```

END WATCH ON ELEMENT RANGE

```
void pend_watch_on_elem_range (

   void

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

13.6 Structure display functions

POST STRUCTURE TO GROUP

```
void ppost_struct_to_grp (
   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* posting group identifier    */
   Pint              struct_id,      /* structure identifier        */
   Pfloat            pri             /* priority*/

);
```

UNPOST STRUCTURES FROM GROUPS

```
void punpost_structs_from_grps (
   Pint              ws_id,          /* workstation identifier      */
   Pint_list         grp_ids,        /* posting group identifiers   */
   Pint_list         struct_ids      /* structure identifiers       */

);
```

UNPOST ALL STRUCTURES FROM GROUP

```
void punpost_all_structs_from_grp (
   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id          /* posting group identifier    */

);
```

POST DIRECT INTERPRETATION STRUCTURE

```
void ppost_di_struct (
   Pint              ws_id,          /* workstation identifier      */
   Pint              struct_id       /* structure identifier        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

POST DIRECT INTERPRETATION STRUCTURE TO POSTING GROUP

```
void ppost_di_struct_to_grp (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* group identifier            */
   Pint              struct_id       /* structure identifier        */

);
```

UNPOST DIRECT INTERPRETATION STRUCTURE

```
void punpost_di_struct (

   Pint              ws_id           /* workstation identifier      */

);
```

RENEW DIRECT INTERPRETATION STATE

```
void prenew_di_state (

   Pint              ws_id,          /* workstation identifier      */
   Pint              struct_id,      /* structure identifier        */
   Ptrav_type        trav_type       /* traversal type              */

);
```

SET DIRECT INTERPRETATION MODE

```
void pset_di_mode (

   Pint              ws_id,          /* workstation identifier      */
   Ptrav_type        trav_type,      /* traversal type              */
   Pdi_mode          mode            /* direct interpretation mode  */

);
```

MARK MULTI-PASS START

```
void pmark_multi_pass_start (

   Pint              ws_id           /* workstation identifier      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

MARK MULTI-PASS COMPLETION

```
void pmark_multi_pass_compl (
   Pint                 ws_id            /* workstation identifier*/
);
```

MARK PASS START

```
void pmark_pass_start (
   Pint                 ws_id            /* workstation identifier      */
);
```

MARK PASS COMPLETION

```
void pmark_pass_compl (
   Pint                 ws_id            /* workstation identifier      */
);
```

RETRIEVE NUMBER OF PASSES REQUIRED

```
void pret_num_passes_req (
   Pint          ws_id,              /* workstation identifier         */
   Ptrav_type    trav_type,          /* traversal type                 */
   Pint          *num_passes_req,    /* OUT picture passes required    */
   Pboolean      *delineation_req    /* OUT pass delineations required */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.7 Input functions

13.7.1 Pick support functions

SET SET PICK FILTER

```
void pset_set_pick_filter (

   Pint                ws_id,          /* workstation identifier      */
   Pint                dev,            /* set dev. number             */
   const Pfilter       *filter         /* echo switch                 */

);
```

SET COMPOSITE PICK FILTER

```
void pset_composite_pick_filter (

   Pint                ws_id,          /* workstation identifier      */
   Pint                dev,            /* composite dev. number       */
   const Pfilter       *filter         /* echo switch                 */

);
```

SET DIRECT INTERPRETATION PICK FILTER

```
void pset_di_pick_filter (

   Pint                ws_id,          /* workstation identifier      */
   const Pfilter       *filter         /* pick filter                 */

);
```

SET DIRECT INTERPRETATION PICK CORRELATION POINT 3

```
void pset_di_pick_corr_point3 (

   Pint                ws_id,          /* workstation identifier      */
   const Ppoint3       *point          /* pick correlation point      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET DIRECT INTERPRETATION PICK CORRELATION POINT

```
void pset_di_pick_corr_point (

   Pint              ws_id,          /* workstation identifier      */
   const Ppoint      *point          /* pick correlation point      */

);
```

SET PICK MAPPING DATA

```
void pset_pick_mapping_data (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 pick_type,   /* pick type                   */
   Ppath_order          order,       /* pick path order             */
   const Pfilter        filter,      /* pick filter                 */
   Pecho_switch         echo,        /* echo switch                 */
   Pint                 pet,         /* prompt and echo type        */
   const Pdi_pick_data  *pick_data   /* pick data record            */

);
```

MAP DEVICE COORDINATE POINT TO PICK PATHS

```
void pmap_dc_point_to_pick_paths (

   Pint              ws_id,          /* workstation identifier      */
   const Ppoint      *point,         /* device coordinate point     */
   Pint              max_paths,      /* max paths to return         */
   Pint              depth,          /* pick path depth             */
   const Ppick_path  *start_path,    /* starting pick path          */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Pin_status        *in_status,     /* OUT pick status             */
   Ppick_path_list   **pick          /* OUT list of pick paths      */

);
```

NOTE — The memory referenced by `*pick` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.7.2 Logical input device definition functions

DEFINE LOCATOR

```
void pdefine_locator (

   Pint                       ws_id,             /* workstation id       */
   Pint                       dev,               /* locator dev. number  */
   Pint                       measure_proc_id,   /* locator measure proc.*/
   const Ptrig_process_list *triggers,           /* list of trigger proc.*/
   const Pecho_process_list *echos,              /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);
```

DEFINE STROKE

```
void pdefine_stroke (

   Pint                       ws_id,             /* workstation id       */
   Pint                       dev,               /* stroke dev. number   */
   Pint                       measure_proc_id,   /* stroke measure proc. */
   const Ptrig_process_list *triggers,           /* list of trigger proc.*/
   const Pecho_process_list *echos,              /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);
```

DEFINE VALUATOR

```
void pdefine_valuator (

   Pint                       ws_id,             /* workstation id.      */
   Pint                       dev,               /* valuator dev. number */
   Pint                       measure_proc_id,   /* valuator measure proc*/
   const Ptrig_process_list *triggers,           /* list of trigger proc.*/
   const Pecho_process_list *echos,              /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

DEFINE CHOICE

```
void pdefine_choice (
   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* choice dev. number   */
   Pint                      measure_proc_id,  /* choice measure proc. */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.*/

);
```

DEFINE PICK

```
void pdefine_pick (
   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* pick dev. number     */
   Pint                      measure_proc_id,  /* pick measure proc.   */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.   */

);
```

DEFINE STRING

```
void pdefine_string (
   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* string dev. number   */
   Pint                      measure_proc_id,  /* string measure proc. */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

CREATE SET MEASURE

```
void pcreate_set_measure (

  Pint                  ws_id,              /* workstation identifier    */
  const Pmeas_process   *base_measure_proc, /* measure process for set   */
  Pint                  max_measures,       /* maximum measures in set   */
  Pint                  *measure_proc_id    /* OUT set measure process   */
                                            /* identifier                */

);
```

DEFINE SET

```
void pdefine_set (

   Pint                     ws_id,             /* workstation id.       */
   Pint                     dev,               /* set dev. number       */
   Pint                     measure_proc_id,   /* set measure proc.     */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.    */
   const Pack_process_list  *acknowledgements /* list of ack. proc.*/

);
```

CREATE COMPOSITE MEASURE

```
void pcreate_composite_measure (

   Pint                     ws_id,           /* workstation identifier */
   const Pmeas_process_list *measure_procs,  /* list of measure proc.  */
   Pint                     *measure_proc_id /* OUT composite measure  */
                                             /* process identifier     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

DEFINE COMPOSITE

```
void pdefine_composite (
   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* composite dev. number*/
   Pint                      measure_proc_id,  /* composite meas. proc.*/
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */
);
```

UNDEFINE LOCATOR

```
void pundefine_locator (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* locator dev. number           */
);
```

UNDEFINE STROKE

```
void pundefine_stroke (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* stroke dev. number            */
);
```

UNDEFINE VALUATOR

```
void pundefine_valuator (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* locator dev. number           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

UNDEFINE CHOICE

```
void pundefine_choice (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* choice dev. number            */
);
```

UNDEFINE PICK

```
void pundefine_pick (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* pick dev. number              */
);
```

UNDEFINE STRING

```
void pundefine_string (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* string dev. number            */
);
```

DESTROY SET MEASURE

```
void pdestroy_set_measure (
  Pint              ws_id,          /* workstation identifier        */
  Pint              measure_proc_id /* set measure process identifier */
);
```

UNDEFINE SET

```
void pundefine_set (
  Pint              ws_id,          /* workstation identifier        */
  Pint              dev             /* set dev. number               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

DESTROY COMPOSITE MEASURE

```
void pdestroy_composite_measure (

  Pint              ws_id,           /* workstation identifier         */
  Pint              measure_proc_id  /* composite measure process id.  */

);
```

UNDEFINE COMPOSITE

```
void pundefine_composite (

  Pint              ws_id,           /* workstation identifier         */
  Pint              dev              /* composite dev. number          */

);
```

13.7.3 Initialization of input devices

INITIALIZE SET 3

```
void pinit_set3 (

   Pint                  ws_id,          /* workstation identifier   */
   Pint                  dev,            /* set dev. number          */
   const Pset_measure    *init_measure,  /* initial measure          */
   Pint                  pet,            /* prompt and echo type     */
   const Plimit3         *echo_volume,   /* echo volume boundaries   */
   const Pset_data       *record         /* input data record        */

);
```

INITIALIZE SET

```
void pinit_set (

   Pint                  ws_id,          /* workstation identifier   */
   Pint                  dev,            /* set dev. number          */
   const Pset_measure    *init_measure,  /* initial measure          */
   Pint                  pet,            /* prompt and echo type     */
   const Plimit          *echo_area,     /* echo area boundaries     */
   const Pset_data       *record         /* input data record */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

INITIALIZE COMPOSITE 3

```
void pinit_composite3 (

   Pint                      ws_id,          /* workstation identifier  */
   Pint                      dev,            /* composite dev. number   */
   const Pcomposite_measure *init_measure,   /* initial measure         */
   Pint                      pet,            /* prompt and echo type    */
   const Plimit3             *echo_volume,   /* echo volume boundaries  */
   const Pcomposite_data     *record         /* input data record       */

);
```

INITIALIZE COMPOSITE

```
void pinit_composite (

   Pint                      ws_id,          /* workstation identifier  */
   Pint                      dev,            /* composite dev. number   */
   const Pcomposite_measure *init_measure,   /* initial measure         */
   Pint                      pet,            /* prompt and echo type    */
   const Plimit              *echo_area,     /* echo area boundaries    */
   const Pcomposite_data     *record         /* input data record       */

);
```

13.7.4 Setting the mode of input devices

SET SET MODE

```
void pset_set_mode (

   Pint              ws_id,          /* workstation identifier      */
   Pint              dev,            /* set dev. number             */
   Pop_mode          mode,           /* operating mode              */
   Pecho_switch      echo            /* echo switch                 */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET COMPOSITE MODE

```
void pset_composite_mode (

   Pint              ws_id,          /* workstation identifier      */
   Pint              dev,            /* composite dev. number       */
   Pop_mode          mode,           /* operating mode              */
   Pecho_switch      echo            /* echo switch                 */

);
```

13.7.5 Request input functions

REQUEST SET 3

```
void preq_set3 (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

REQUEST SET

```
void preq_set (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

REQUEST COMPOSITE 3

```
void preq_composite3 (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

REQUEST COMPOSITE

```
void preq_composite (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

13.7.6 Sample input functions

SAMPLE SET 3

```
void psample_set3 (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SAMPLE SET

```
void psample_set (
   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by *measure is managed by store.

SAMPLE COMPOSITE 3

```
void psample_composite3 (
   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by *measure is managed by store.

SAMPLE COMPOSITE

```
void psample_composite (
   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by *measure is managed by store..

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.7.7 Event input functions

GET SET 3

```
void pget_set3 (
   Pstore                  store,      /* store for measure data      */
   Pna_in_status           *status,    /* OUT non-atomic input status */
   Pset_measure            **measure   /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

GET SET

```
void pget_set (
   Pstore                  store,      /* store for measure data      */
   Pna_in_status           *status,    /* OUT non-atomic input status */
   Pset_measure            **measure   /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

GET COMPOSITE 3

```
void pget_composite3 (
   Pstore                  store,      /* store for measure data      */
   Pna_in_status           *status,    /* OUT non-atomic input status */
   Pcomposite_measure      **measure   /* OUT pointer to measure data */
);
```

NOTE — The memory referenced by `*measure` is managed by `store`

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

GET COMPOSITE

```
void pget_composite (

   Pstore                  store,      /* store for measure data      */
   Pna_in_status           *status,    /* OUT non-atomic input status */
   Pcomposite_measure      **measure   /* OUT pointer to measure data */

);
```

NOTE — The memory referenced by `*measure` is managed by `store`.

13.7.8 Local input operation functions

ATTACH LOGICAL INPUT DEVICE TO VIEW

```
void pattach_lid_to_view (

   Pint            lid_ws_id,          /* LID workstation id          */
   Pin_class       dev_class,          /* LID class                   */
   Pint            dev,                /* LID number                  */
   Pint            view_ws_id,         /* view workstation id         */
   Pint            view_ind,           /* view index                  */
   Pint            view_action_type,   /* action type                 */
   Pint            transfer_type,      /* measure transfer type       */
   Pint            view_upd_method     /* view update method          */

);
```

DETACH LOGICAL INPUT DEVICE FROM VIEW

```
void pdetach_lid_from_view (

   Pint            lid_ws_id,          /* LID workstation id          */
   Pin_class       dev_class,          /* LID class                   */
   Pint            dev,                /* LID number                  */
   Pint            view_ws_id,         /* view workstation id         */
   Pint            view_ind,           /* view index                  */
   Pint            view_action_type,   /* view action type            */
   Pint            view_upd_method     /* view update method          */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.7.9 Local lighting functions

ATTACH LOGICAL INPUT DEVICE TO LIGHT SOURCE

```
void pattach_lid_to_light_src (

   Pint          lid_ws_id,          /* LID workstation id          */
   Pin_class     dev_class,          /* LID class                   */
   Pint          dev,                /* LID number                  */
   Pint          light_ws_id,        /* light ws id                 */
   Pint          light_src_ind,      /* light source index          */
   Pint          action,             /* light action type           */
   Pint          transfer_type       /* measure transfer type       */

);
```

DETACH LOGICAL INPUT DEVICE FROM LIGHT SOURCE

```
void pdetach_lid_from_light_src (

   Pint          LID_ws_id,          /* LID workstation id          */
   Pin_class     dev_class,          /* LID class                   */
   Pint          in_num,             /* LID number                  */
   Pint          light_ws_id,        /* light ws id                 */
   Pint          light_src_ind,      /* light source index          */
   Pint          action              /* light action type           */

);
```

13.7.10 Direct interpretation picking functions

INITIALIZE DIRECT INTERPRETATION PICK 3

```
void pinit_di_pick3 (

   Pint                  ws_id,        /* workstation identifier     */
   Pint                  pick_type,    /* pick type                  */
   Pint                  max_sets,     /* max. pick sets to return   */
   Pecho_switch          echo_switch,  /* echo switch                */
   Pint                  pet,          /* prompt and echo type       */
   const Plimit3         *echo_vol,    /* echo volume                */
   const Pdi_pick_data3  *pick_data,   /* pick data record           */
   Ppath_order           order         /* pick path order            */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INITIALIZE DIRECT INTERPRETATION PICK

```
void pinit_di_pick (

   Pint                 ws_id,        /* workstation identifier      */
   Pint                 pick_type,    /* pick type                   */
   Pint                 max_sets,     /* max. pick sets to return    */
   Pecho_switch         echo_switch,  /* echo switch                 */
   Pint                 pet,          /* prompt and echo type        */
   const Plimit         *echo_area,   /* echo area                   */
   const Pdi_pick_data  *pick_data,   /* pick data record            */
   Ppath_order          order         /* pick path order             */

);
```

ENABLE DIRECT INTERPRETATION PICK

```
void penable_di_pick (

   Pint                 ws_id,        /* workstation identifier      */
   Pint                 struct_id     /* structure identifier        */

);
```

DISABLE DIRECT INTERPRETATION PICK

```
void pdisable_di_pick (

   Pint                 ws_id         /* workstation identifier      */

);
```

TRANSFER DIRECT INTERPRETATION PICK SET

```
void ptrans_di_pick_set (

   Pint                 ws_id,        /* workstation identifier      */
   Pstore               store,        /* handle to Store object      */
   Pin_status           *in_status,   /* OUT pick status             */
   Ppick_path_list      **pick        /* OUT direct interpretation   */
                                      /* pick sets                   */

);
```

NOTE — The memory referenced by `*pick` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.8 Inquiry functions

13.8.1 Inquiry functions for PHIGS description table

INQUIRE LINETYPE DEFINITION FACILITIES

```
void pinq_linetype_def_facs (

   Pstore              store,          /* handle to store object        */
   Pint                *err_ind,       /* OUT error indicator           */
   Plinetype_def_facs  **facs          /* OUT linetype definition       */
                                       /* facilities                    */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE PREDEFINED LINETYPE DEFINITION

```
void pinq_pred_linetype_def (

   Pint         linetype,              /* linetype                      */
   Pint         num_elems_appl_list,   /* # elems in appl list          */
   Pint         starting_ind,          /* starting element to return    */
   Pint         *err_ind,              /* OUT error indicator           */
   Pfloat       *repeat_length,        /* OUT dash cycle repeat length
                                          */
   Pfloat_list  *segment_lengths,      /* OUT list of segment lengths   */
   Pint         *num_elems_impl_list   /* OUT # elems in impl list      */

);
```

INQUIRE MARKER TYPE DEFINITION FACILITIES

```
void pinq_marker_type_def_facs (

   Pstore                  store,      /* handle to store object        */
   Pint                    *err_ind,   /* OUT error indicator           */
   Pmarker_type_def_facs   **facs      /* OUT marker type definition    */
                                       /* facilities                    */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED MARKER TYPE DEFINITION

```
void pinq_pred_marker_type_def (

   Pint           marker_type,        /* marker type                  */
   Pstore         store,              /* handle to store object       */
   Pint           *err_ind,           /* OUT error indicator          */
   Pmarker_desc   **marker            /* OUT predef. marker descriptors
                                         */

);
```

NOTE — The memory referenced by `*marker` is managed by `store`..

13.8.2 Inquiry functions for PHIGS state list

INQUIRE SET OF WORKSTATIONS TO WHICH DIRECT INTERPRETATION STRUCTURE POSTED

```
void pinq_wss_di_struct_posted (

   Pint       num_elems_appl_list, /* # elems in appl. list          */
   Pint       starting_ind,        /* starting element to return     */
   Pint       *err_ind,            /* OUT error indicator            */
   Pint_list  *ws_id,              /* OUT list of ws identifiers     */
   Pint       *num_elems_impl_list /* OUT # elems in impl list       */

);
```

INQUIRE WATCH ON ELEMENT RANGE

```
void pinq_watch_elem_range (

   Pint     *err_ind,              /* OUT error indicator            */
   Poff_on  *watch_enable_flag,    /* OUT watch enable flag          */
   Pint     *struct_id,            /* OUT current watched structure  */
   Pint     *elem_pos1,            /* OUT element position 1         */
   Pint     *elem_pos2             /* OUT element position 2         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE WATCH RANGE STATE

```
void pinq_watch_range_st (

   Pint     *err_ind,                /* OUT error indicator              */
   Poff_on  *primitive_erase_req,    /* OUT primitive erasures required */
   Poff_on  *primitive_redraw_req    /* OUT primitive redraws required  */

);
```

INQUIRE NUMBER OF DEFINED LINETYPES

```
void pinq_num_def_linetypes (

   Pint  *err_ind,              /* OUT error indicator                */
   Pint  *num_defined           /* OUT number defined linetypes       */

);
```

INQUIRE NUMBER OF DEFINED MARKER TYPES

```
void pinq_num_def_marker_types (

   Pint  *err_ind,              /* OUT error indicator                */
   Pint  *num_defined           /* OUT number defined marker types    */

);
```

INQUIRE LINETYPE DEFINITION

```
void pinq_linetype_def (

   Pint         linetype,              /* linetype                     */
   Pint         num_elems_appl_list,   /* # elems in appl list         */
   Pint         starting_ind,          /* starting element to return   */
   Pint         *err_ind,              /* OUT error indicator          */
   Pfloat       *repeat_length,        /* OUT dash cycle repeat length*/
   Pfloat_list  *segment_lengths,      /* OUT list of segment lengths */
   Pint         *num_elems_impl_list   /* OUT # elems in impl list     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE MARKER TYPE DEFINITION

```
void pinq_marker_type_def (

   Pint           marker_type,        /* marker type                  */
   Pstore         store,              /* handle to store object       */
   Pint           *err_ind,           /* OUT error indicator          */
   Pmarker_desc   **marker            /* OUT list of marker descriptors
                                         */

);
```

NOTE — The memory referenced by `*marker` is managed by `store`.

13.8.3 Inquiry functions for workstation state list

INQUIRE LIST OF POSTING GROUPS

```
void pinq_list_post_grps (

   Pint       ws_id,                 /* workstation identifier       */
   Pint       num_elems_appl_list,   /* # elems in appl. list        */
   Pint       starting_ind,          /* starting elem to return      */
   Pint       *err_ind,              /* OUT error indicator          */
   Pint_list  *grp_ids,              /* OUT list of posting groups   */
   Pint       *num_elems_impl_list   /* OUT # elems in impl. list    */

);
```

INQUIRE ASSOCIATED IMAGE RESOURCES

```
void pinq_assoc_image_res (

   Pint       ws_id,                 /* workstation identifier       */
   Pint       num_elems_appl_list,   /* # elems in appl list         */
   Pint       starting_ind,          /* start index                  */
   Pint       *err_ind,              /* OUT error indicator          */
   Pint_list  **image_res_ids,       /* OUT list of image resource   */
                                     /* identifiers                  */
   Pint       *num_elems_impl_list   /* OUT # elems in impl list     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DEVICE COORDINATE CLIP REGIONS 3

```
void pinq_dc_clip_regions3 (

   Pint                ws_id,          /* workstation identifier      */
   Pstore              store,          /* handle to Store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Plimit3_list        **limit_list    /* OUT list of clip regions    */

);
```

NOTE — The memory referenced by `*limit_list` is managed by `store`.

INQUIRE DEVICE COORDINATE CLIP REGIONS

```
void pinq_dc_clip_regions (

   Pint                ws_id,          /* workstation identifier      */
   Pstore              store,          /* handle to Store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Plimit_list         **limit_list    /* OUT list of clip regions    */

);
```

NOTE — The memory referenced by `*limit_list` is managed by `store`.

INQUIRE TARGET MANIPULATION MODE

```
void pinq_targ_manip_mode (

   Pint           ws_id,               /* workstation identifier      */
   Pint           *err_ind,            /* OUT error indicator         */
   Pint           *targ_manip_mode     /* OUT target manipulation mode*/

);
```

INQUIRE TARGET DISPOSITION

```
void pinq_targ_dispos (

   Pint                ws_id,          /* workstation identifier      */
   Pstore              store,          /* handle to Store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Ptarg_op_list       **targ_ops      /* OUT target operations list  */

);
```

NOTE — The memory referenced by `*targ_ops` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DISPLAY TARGET

```
void pinq_disp_targ (
   Pint              ws_id,          /* workstation identifier      */
   Pint              *err_ind,       /* OUT error indicator         */
   Pint              *offset         /* OUT offset from base target */
);
```

INQUIRE RENDERING TARGET

```
void pinq_rend_targ (
   Pint              ws_id,          /* workstation identifier      */
   Pint              *err_ind,       /* OUT error indicator         */
   Pint              *offset         /* OUT offset from base target */
);
```

INQUIRE LIST OF HIGHLIGHTING INDICES

```
void pinq_list_highl_inds (
   Pint        ws_id,                /* workstation identifier      */
   Pint        num_elems_appl_list,  /* # elems in appl list        */
   Pint        starting_ind,         /* starting element to return  */
   Pint        *err_ind,             /* OUT error indicator         */
   Pint_list   *highl_inds,          /* OUT list of highl indices   */
   Pint        *num_elems_impl_list  /* OUT # elems in impl list    */
);
```

INQUIRE LIST OF DEFINED APPLICATION FILTERS

```
void pinq_list_def_appl_filters (
   Pint        ws_id,                /* workstation id              */
   Pint        num_elems_appl_list,  /* num elems in appl list      */
   Pint        start_ind,            /* start elem index            */
   Pint        *err_ind,             /* OUT error indicator         */
   Pint_list   *filter_ids,          /* OUT list of appl. filters   */
   Pint        *num_elems_impl_list  /* OUT num elems in impl list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LIST OF TEXTURE INDICES

```
void pinq_list_texture_inds (

   Pint         ws_id,                  /* workstation id                */
   Pint         num_elems_appl_list,    /* num elems in appl list        */
   Pint         start_ind,              /* start elem index              */
   Pint         *err_ind,               /* OUT error indicator           */
   Pint_list    *indices,               /* OUT list of defined texture */
                                        /* indices                       */
   Pint         *num_elems_impl_list    /* OUT num elems in impl list  */

);
```

INQUIRE TRANSPARENCY MODE

```
void pinq_transparency_mode (

   Pint                 ws_id,          /* workstation id                */
   Pint                 *err_ind,       /* OUT error indicator           */
   Pint                 *mode           /* OUT transparency mode         */

);
```

INQUIRE TRANSPARENCY THRESHOLDS

```
void pinq_transparency_thresholds (

   Pint         ws_id,                  /* workstation id                */
   Pint         num_elems_appl_list,    /* num elems in appl list        */
   Pint         start_ind,              /* start elem index              */
   Pint         *err_ind,               /* OUT error indicator           */
   Pfloat_list  *thresholds,            /* OUT list of thresholds        */
   Pint         *num_elems_impl_list    /* OUT num elems in impl list  */

);
```

INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 3

```
void pinq_di_pick_corr_point3 (

   Pint                 ws_id,          /* workstation identifier        */
   Pint                 *err_ind,       /* OUT error indicator           */
   Ppoint3              *point          /* OUT pick correlation point  */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT

```
void pinq_di_pick_corr_point (

   Pint              ws_id,          /* workstation identifier      */
   Pint              *err_ind,       /* OUT error indicator         */
   Ppoint            *point          /* OUT pick correlation point  */

);
```

INQUIRE DIRECT INTERPRETATION PICK SET STATUS

```
void pinq_di_pick_set_status (

   Pint              ws_id,          /* workstation identifier      */
   Pint              *err_ind,       /* OUT error indicator         */
   Ppick_stat        *pick_stat      /* OUT maximum number of pick  */
                                     /* reports reached             */

);
```

INQUIRE PICK MAPPING STATE

```
void pinq_pick_mapping_state (

   Pint              ws_id,          /* workstation identifier      */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Pecho_switch      *echo,          /* OUT pick echo switch        */
   Pfilter           **filter,       /* OUT pick filter             */
   Pint              *pick_type,     /* OUT pick type               */
   Pint              *pet,           /* OUT pick mapping prompt and */
                                     /* echo type                   */
   Pdi_pick_data     **pick_data,    /* OUT pick data record        */
   Ppath_order       *order          /* pick path order             */

);
```

NOTE — The memory referenced by `*filter`, `*pick_type`, and `*pick_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LISTS OF AVAILABLE LOGICAL INPUT DEVICES

```
void pinq_lists_avail_lids (

   Pint        ws_id,              /* workstation identifier           */
   Pstore      store,              /* handle to Store object           */
   Pint        *err_ind,           /* OUT error indicator              */
   Pint_list   **avail_loc_dev,    /* OUT list available locator LIDs  */
   Pint_list   **avail_stroke_dev,/* OUT list available stroke LIDs    */
   Pint_list   **avail_val_dev,    /* OUT list available valuator LIDs */
   Pint_list   **avail_choice_dev,/* OUT list available choice LIDs    */
   Pint_list   **avail_pick_dev,   /* OUT list available pick LIDs     */
   Pint_list   **avail_string_dev,/* OUT list available string LIDs    */
   Pint_list   **avail_set_dev,    /* OUT list available set LIDs      */
   Pint_list   **avail_compos_dev /* OUT list available composite LIDs */

);
```

INQUIRE POSTING GROUP

```
void pinq_post_grp (

   Pint              ws_id,            /* workstation identifier      */
   Pint              grp_id,           /* posting group identifier    */
   Pstore            store,            /* handle to Store object      */
   Pint              *err_ind,         /* OUT error indicator         */
   Pint              *view_ind,        /* OUT default view index      */
   Ppost_grp_status  *post_grp_status, /* OUT posting group status    */
   Pbackg_style      *backg_style,     /* OUT background style        */
   Pbackg_method     **backg_method,   /* OUT background method data  */
   Pborder_indic     *border_indic,    /* OUT border indicator        */
   Pint              *border_ind       /* OUT border index            */

);
```

NOTE — The memory referenced by `*backg_method` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE POSTED STRUCTURES FROM POSTING GROUP

```
void pinq_posted_structs_from_post_grp (

   Pint       ws_id,                /* workstation id.                  */
   Pint       num_elems_appl_list,  /* # elems in appl list             */
   Pint       start_ind,            /* starting index                   */
   Pint       *err_ind,             /* OUT error indicator              */
   Pint_list  *struct_ids,          /* OUT list of posted struct nets   */
   Pint       *num_elems_impl_list  /* OUT # elems in impl list         */

);
```

INQUIRE DIRECT INTERPRETATION MODE

```
void pinq_di_mode (

   Pint              ws_id,         /* workstation identifier          */
   Ptrav_type        trav_type,     /* traversal type                  */
   Pint              *err_ind,      /* OUT error indicator             */
   Pdi_mode          *mode          /* OUT direct interpretation mode  */

);
```

INQUIRE PICTURE STATUS

```
void pinq_pict_status (

   Pint              ws_id,          /* workstation identifier       */
   Ptrav_type        trav_type,      /* traversal type               */
   Pint              *err_ind,       /* OUT error indicator          */
   Ppict_stat        *pict_stat      /* OUT picture status           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TARGET STATE

```
void pinq_targ_st (

   Pint               ws_id,          /* workstation identifier      */
   const Ptarg_addr   *targ_addr,     /* target address              */
   Pstore             store,          /* handle to Store object      */
   Pint               *err_ind,       /* OUT error indicator         */
   Pint               *offset,        /* OUT offset from base target */
   Ptarg_data_st      *st_nr_pict,    /* OUT state of non-retained   */
                                      /* picture                     */
   Ptarg_data_st      *st_ret_pict,   /* OUT state of retained       */
                                      /* picture                     */
   Pvisual_st         *visual_st,     /* OUT target state of         */
                                      /* visual rep.                 */
   Ptarg_empty_st     *empty_st,      /* OUT target empty            */
   Ptarg_type_list    **targ_types    /* OUT target usage list       */

);
```

NOTE — The memory referenced by `*targ_types` is managed by `store`.

INQUIRE TRAVERSAL RESOURCES ASSOCIATED WITH TARGET

```
void pinq_trav_res_assoc_targ (

   Pint               ws_id,                /* workstation identifier   */
   const Ptarg_addr   *targ_addr,           /* target address           */
   Pint               num_elems_appl_list,  /* # elems in appl list     */
   Pint               starting_ind,         .* start index             */
   Pint               *err_ind,             /* OUT error indicator      */
   Pint_list          *res_ids,             /* OUT list of associated   */
                                            /* traversal resources      */
   Pint               *num_elems_impl_list  /* OUT # elems in impl list */

);
```

INQUIRE TARGET ASSOCIATED WITH TRAVERSAL RESOURCE

```
void pinq_targ_assoc_trav_res (

   Pint               ws_id,          /* workstation identifier         */
   Pint               res_id,         /* traversal resource identifier  */
   Pint               *err_ind,       /* OUT error indicator            */
   Passoc_flag        *assoc_flag,    /* OUT association flag           */
   Pint               *offset         /* OUT offset from base target    */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE IMAGE RESOURCE

```
void pinq_image_res (

   Pint              ws_id,          /* workstation identifier      */
   Pint              image_res_id,   /* image resource identifier   */
   Pstore            store,          /* store                       */
   Pint              *err_ind,       /* OUT error indicator         */
   Pint_size         *dims,          /* OUT m,n dimensions          */
   Pimage_res        **image_res,    /* OUT image resource          */
   Paccess_flag      *access_flag    /* OUT access flag             */

);
```

NOTE — The memory referenced by `*image_res` is managed by `store`.

INQUIRE POSTED DIRECT INTERPRETATION STRUCTURE

```
void pinq_posted_di_struct (

   Pint              ws_id,          /* workstation identifier      */
   Ptrav_type        trav_type,      /* traversal type              */
   Pint              *err_ind,       /* OUT error indicator         */
   Pposted_ind       *posted_ind,    /* OUT posted indicator        */
   Pint              *grp_id         /* OUT posting group identifier*/

);
```

INQUIRE POLYLINE REPRESENTATION FULL

```
void pinq_line_rep_full (

   Pint              ws_id,          /* workstation id              */
   Pint              index,          /* polyline bundle index       */
   Pint              *err_ind,       /* OUT error indicator         */
   Pline_bundle_full *rep            /* polyline representation full */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE EDGE REPRESENTATION FULL

```
void pinq_edge_rep_full (

   Pint                ws_id,          /* workstation id                */
   Pint                index,          /* edge bundle index             */
   Pint                *err_ind,       /* OUT error indicator           */
   Pedge_bundle_full   *rep            /* edge representation full      */

);
```

INQUIRE EXTENDED PATTERN REPRESENTATION

```
void pinq_ext_pat_rep (

   Pint                ws_id,          /* workstation identifier        */
   Pint                index,          /* pattern index                 */
   Pinq_type           type,           /* type of returned value        */
   Pstore              store,          /* handle to store object        */
   Pint                *err_ind,       /* OUT error indicator           */
   Ppattern            **pattern       /* OUT pattern data              */

);
```

NOTE — The memory referenced by `*pattern` is managed by `store`.

INQUIRE VIEW STATUS

```
void pinq_view_status (

   Pint                ws_id,          /* workstation id                */
   Pint                view_ind,       /* view index                    */
   Pint                *err_ind,       /* OUT error indicator           */
   Pview_status        *view_status    /* OUT view status               */

);
```

INQUIRE VIEW REFERENCE POINT 3

```
void pinq_view_ref_point3 (

   Pint              ws_id,            /* workstation id                */
   Pint              view_ind,         /* view index                    */
   Pint              *err_ind,         /* OUT error indicator           */
   Ppoint3           *view_ref_point   /* OUT view reference point      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE VIEW REFERENCE POINT

```
void pinq_view_ref_point (

   Pint              ws_id,              /* workstation id              */
   Pint              view_ind,           /* view index                  */
   Pint              *err_ind,           /* OUT error indicator         */
   Ppoint            *view_ref_point     /* OUT view reference point    */

);
```

INQUIRE VIEW PLANE NORMAL

```
void pinq_view_plane_norm (

   Pint              ws_id,              /* workstation id              */
   Pint              view_ind,           /* view index                  */
   Pint              *err_ind,           /* OUT error indicator         */
   Pvec3             *view_norm_vec      /* OUT view plane normal       */

);
```

INQUIRE VIEW UP VECTOR 3

```
void pinq_view_up_vec3 (

   Pint              ws_id,              /* workstation id              */
   Pint              view_ind,           /* view index                  */
   Pint              *err_ind,           /* OUT error indicator         */
   Pvec3             *view_up_vec        /* OUT view up vector          */

);
```

INQUIRE VIEW UP VECTOR

```
void pinq_view_up_vec (

   Pint              ws_id,              /* workstation id              */
   Pint              view_ind,           /* view index                  */
   Pint              *err_ind,           /* OUT error indicator         */
   Pvec              *view_up_vec        /* OUT view up vector          */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE VIEW WINDOW LIMITS

```
void pinq_view_win_limits (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Plimit            *win            /* OUT view window limits      */

);
```

INQUIRE PROJECTION VIEWPORT 3

```
void pinq_proj_vp3 (

   Pint           ws_id,          /* workstation id                 */
   Pint           view_ind,       /* view index                     */
   Pint           *err_ind,       /* OUT error indicator            */
   Plimit3        *proj_vp        /* OUT projection viewport limits */

);
```

INQUIRE PROJECTION VIEWPORT

```
void pinq_proj_vp (

   Pint            ws_id,          /* workstation id                 */
   Pint            view_ind,       /* view index                     */
   Pint            *err_ind,       /* OUT error indicator            */
   Plimit          *proj_vp        /* OUT projection viewport limits */

);
```

INQUIRE PROJECTION TYPE

```
void pinq_proj_type (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pproj_type        *proj_type      /* OUT projection type         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PROJECTION REFERENCE POINT

```
void pinq_proj_ref_point (

   Pint          ws_id,              /* workstation id                */
   Pint          view_ind,           /* view index                    */
   Pint          *err_ind,           /* OUT error indicator           */
   Ppoint3       *proj_ref_point     /* OUT projection reference point */

);
```

INQUIRE VIEW PLANE DISTANCE

```
void pinq_view_plane_dist (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   Pint              *err_ind,       /* OUT error indicator           */
   Pfloat            *view_plane     /* OUT view plane distance       */

);
```

INQUIRE FRONT PLANE DISTANCE

```
void pinq_front_plane_dist (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   Pint              *err_ind,       /* OUT error indicator           */
   Pfloat            *front_plane    /* OUT front plane distance      */

);
```

INQUIRE BACK PLANE DISTANCE

```
void pinq_back_plane_dist (

   Pint              ws_id,          /* workstation id                */
   Pint              view_ind,       /* view index                    */
   Pint              *err_ind,       /* OUT error indicator           */
   Pfloat            *back_plane     /* OUT back plane distance       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE X-Y CLIPPING INDICATOR

```
void pinq_xy_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *xy_clip        /* OUT x-y clipping indicator  */

);
```

INQUIRE FRONT CLIPPING INDICATOR

```
void pinq_front_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *front_clip     /* OUT front clipping indicator*/

);
```

INQUIRE BACK CLIPPING INDICATOR

```
void pinq_back_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *back_clip      /* OUT back clipping indicator */

);
```

INQUIRE HIGHLIGHTING REPRESENTATION

```
void pinq_highl_rep (

   Pint              ws_id,          /* workstation identifier      */
   Pint              index,          /* highlighting index          */
   Pinq_type         type,           /* type of returned value      */
   Pstore            store,          /* handle to store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Phighl_method     **method        /* OUT highlighting method data*/

);
```

NOTE — The memory referenced by `*method` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TEXTURE REPRESENTATION

```
void pinq_texture_rep (

   Pint              ws_id,          /* workstation id              */
   Pint              index,          /* texture index               */
   Pinq_type         type,           /* type of returned values     */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Ptexture_rep      **rep           /* OUT texture representation  */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE DIRECT INTERPRETATION PICK STATE 3

```
void pinq_di_pick_st3 (

   Pint              ws_id,              /* workstation identifier     */
   Pinq_type         type,               /* type of returned value     */
   Pstore            store,              /* store for measure data     */
   Pint              *err_ind,           /* OUT error indicator        */
   Pecho_switch      *echo,              /* OUT echo switch            */
   Pfilter           **pick_filter,      /* OUT di pick filter         */
   Ppoint3           *correlation_pt,    /* OUT pick correlation point */
   Pint              *pick_type,         /* OUT pick type              */
   Pint              *max_sets,          /* OUT max sets to return     */
   Pin_status        *init_status,       /* OUT initial status         */
   Ppick_path_list   **init_pick_sets,   /* OUT initial di pick sets   */
   Pint              *pet,               /* OUT prompt and echo type   */
   Plimit3           *echo_vol,          /* OUT echo volume            */
   Ppick_data        **pick_data,        /* OUT input data record      */
   Ppath_order       *order              /* OUT pick path order        */

);
```

NOTE — The memory referenced by `*pick_filter`, `*pick_type`, `*init_pick_sets`, and `*pick_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DIRECT INTERPRETATION PICK STATE

```
void pinq_di_pick_st (

   Pint              ws_id,              /* workstation identifier      */
   Pinq_type         type,               /* type of returned value      */
   Pstore            store,              /* store for measure data      */
   Pint              *err_ind,           /* OUT error indicator         */
   Pecho_switch      *echo_switch,       /* OUT echo switch             */
   Pfilter           **pick_filter,      /* OUT di pick filter          */
   Ppoint            *correlation_pt,    /* OUT pick correlation point  */
   Pint              *pick_type,         /* OUT pick type               */
   Pint              *max_sets,          /* OUT max sets to return      */
   Pin_status        *init_status,       /* OUT initial status          */
   Ppick_path_list   **init_pick_sets,   /* OUT initial di pick sets    */
   Pint              *pet,               /* OUT prompt and echo type    */
   Plimit            *echo_area,         /* OUT echo area               */
   Ppick_data        **pick_data,        /* OUT input data record       */
   Ppath_order       *order              /* OUT pick path order         */

);
```

NOTE — The memory referenced by `*pick_filter`, `*pick_type`, `*init_pick_sets`, and `*pick_data` is managed by `store`.

INQUIRE APPLICATION FILTER

```
void pinq_appl_filter (

   Pint              ws_id,              /* workstation identifier      */
   Pint              filter_sel,         /* filter selection            */
   Pstore            store,              /* handle to store object      */
   Pint              *err_ind,           /* OUT error indicator         */
   Pfilter           **appl_filter       /* OUT application filter      */

);
```

NOTE — The memory referenced by `*appl_filter` is managed by `store`.

NOTE — The memory referenced by `*avail_loc_dev`, `*avail_stk_dev`, `*avail_val_dev`, `*avail_cho_dev`, `*avail_pik_dev`, `*avail_stg_dev`, `*avail_set_dev`, and `*avail_cmp_dev` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LOGICAL INPUT DEVICE DEFINITION

```
void pinq_lid_def (

   Pint                 ws_id,          /* workstation identifier   */
   Pin_class            dev_class,      /* input class              */
   Pint                 device,         /* device number            */
   Pstore               store,          /* handle to Store object   */
   Pint                 *err_ind,       /* OUT error indicator      */
   Pint                 *meas_proc,     /* OUT measure process      */
   Ptrig_process_list   **trig_procs,   /* OUT trigger processes    */
   Pecho_process_list   **echo_procs,   /* OUT echo processes       */
   Pack_process_list    **ack_procs     /* OUT ack. processes       */

);
```

NOTE — The memory referenced by `*trig_procs`, `*echo_procs`, and `*ack_procs` is managed by `store`.

INQUIRE SET MEASURE DEFINITION

```
void pinq_set_meas_def (

   Pint                 ws_id,          /* workstation identifier   */
   Pint                 meas_id,        /* set measure process id   */
   Pint                 *err_ind,       /* OUT error indicator      */
   Pint                 *max_meas,      /* OUT max measures in set  */
   Pmeas_process        *meas_proc      /* OUT base measure process */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE SET DEVICE STATE 3

```
void pinq_set_st3 (

   Pint              ws_id,          /* workstation identifier     */
   Pint              dev,            /* set device number          */
   Pstore            store,          /* store for measure data     */
   Pint              *err_ind,       /* OUT error indicator        */
   Pop_mode          *mode,          /* OUT operating mode         */
   Pecho_switch      *echo,          /* OUT echo switch            */
   Pfilter           **filter,       /* OUT set pick filter        */
   Pset_measure      **measure,      /* OUT initial measure        */
   Pint              *pet,           /* OUT prompt and echo type   */
   Plimit3           *echo_vol,      /* OUT echo volume            */
   Pset_data         **set_data      /* OUT input data record      */

);
```

NOTE — The memory referenced by `*filter`, `*measure`, and `*set_data` is managed by `store`.

INQUIRE SET DEVICE STATE

```
void pinq_set_st (

   Pint              ws_id,          /* workstation identifier     */
   Pint              dev,            /* set device number          */
   Pstore            store,          /* store for measure data     */
   Pint              *err_ind,       /* OUT error indicator        */
   Pop_mode          *mode,          /* OUT operating mode         */
   Pecho_switch      *echo,          /* OUT echo switch            */
   Pfilter           **filter,       /* OUT set pick filter        */
   Pset_measure      **measure,      /* OUT initial measure        */
   Pint              *pet,           /* OUT prompt and echo type   */
   Plimit            *echo_area,     /* OUT echo area              */
   Pset_data         **set_data      /* OUT input data record      */

);
```

NOTE — The memory referenced by `*filter`, `*measure`, and `*set_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE COMPOSITE MEASURE DEFINITION

```
void pinq_composite_meas_def (

   Pint                  ws_id,          /* workstation identifier     */
   Pint                  meas_id,        /* composite measure proc id  */
   Pstore                store,          /* store for measure procs    */
   Pint                  *err_ind,       /* OUT error indicator        */
   Pmeas_process_list    **meas_procs    /* OUT measure processes      */

);
```

NOTE — The memory referenced by `*meas_procs` is managed by `store`

INQUIRE COMPOSITE DEVICE STATE 3

```
void pinq_composite_st3 (

   Pint                  ws_id,          /* workstation identifier     */
   Pint                  dev,            /* composite dev. number      */
   Pstore                store,          /* store for measure data     */
   Pint                  *err_ind,       /* OUT error indicator        */
   Pop_mode              *mode,          /* OUT operating mode         */
   Pecho_switch          *echo,          /* OUT echo switch            */
   Pfilter               **filter,       /* OUT composite pick filter  */
   Pcomposite_measure    **measure,      /* OUT initial measure        */
   Pint                  *pet,           /* OUT prompt and echo type   */
   Plimit3               *echo_vol,      /* OUT echo volume            */
   Pcomposite_data       **composite_data /* OUT input data record     */

);
```

NOTE — The memory referenced by `*filter`, `*measure`, and `*composite_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE COMPOSITE DEVICE STATE

```
void pinq_composite_st (

   Pint                ws_id,            /* workstation identifier     */
   Pint                dev,              /* composite dev. number      */
   Pstore              store,            /* store for measure data     */
   Pint                *err_ind,         /* OUT error indicator        */
   Pop_mode            *mode,            /* OUT operating mode         */
   Pecho_switch        *echo,            /* OUT echo switch            */
   Pfilter             **filter,         /* OUT composite pick filter  */
   Pcomposite_measure  **measure,        /* OUT initial measure        */
   Pint                *pet,             /* OUT prompt and echo type   */
   Plimit              *echo_area,       /* OUT echo area              */
   Pcomposite_data     **composite_data  /* OUT input data record      */

);
```

NOTE — The memory referenced by `*filter`, `*measure`, and `*composite_data` is managed by `store`.

INQUIRE LOGICAL INPUT DEVICE ATTACHED TO VIEW

```
void pinq_lid_attached_to_view (

   Pint          ws_id,          /* workstation identifier              */
   Pin_class     dev_class,      /* input class                         */
   Pint          dev,            /* logical input dev. number           */
   Pstore        store,          /* store for list of attachment data   */
   Pint          *err_ind,       /* OUT error indicator                 */
   Pint          *num_attached,  /* OUT num view attachments for LID    */
   Pview_attach  **attached      /* OUT list of view attachments for LID*/

);
```

NOTE — The memory referenced by `*attached` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LOGICAL INPUT DEVICE ATTACHED TO LIGHT SOURCE

```
void pinq_lid_attached_to_light_src (

   Pint          ws_id,          /* workstation identifier            */
   Pin_class     dev_class,      /* input class                       */
   Pint          dev,            /* logical input dev. number         */
   Pstore        store,          /* store for list of attachment data */
   Pint          *err_ind,       /* OUT error indicator               */
   Pint          *num_attached,  /* OUT num light source attachments  */
                                 /* for LID                           */
   Plight_attach **attached      /* OUT list of light source          */
                                 /* attachments for LID*/

);
```

NOTE — The memory referenced by `*attached` is managed by `store`.

13.8.4 Inquiry functions for workstation description table

INQUIRE DEVICE COORDINATE CLIP REGIONS FACILITIES

```
void pinq_dc_clip_regions_facs(

   Pint          ws_type,              /* workstation type            */
   Pstore        store,                /* handle to Store object      */
   Pint          *err_ind,             /* OUT error indicator         */
   Pdc_clip_facs **facs                /* OUT DC clip region facilities
                                          */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE CONFIGURATION SETTING FACILITIES

```
void pinq_config_setting_facs (

   Pint                  ws_type,        /* workstation type        */
   Pstore                store,          /* handle to Store object  */
   Pint                  *err_ind,       /* OUT error indicator     */
   Pconfig_setting_facs  **facs,         /* OUT configuration setting
                                            */
                                         /* facilities              */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TARGET FACILITIES

```
void pinq_targ_facs (

   Pint          ws_type,          /* workstation type                */
   Pstore        store,            /* handle to Store object          */
   Pint          *err_ind,         /* OUT error indicator             */
   Ptarget_facs  *facs             /* OUT target facilities           */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE DEFAULT TARGET DISPOSITION

```
void pinq_def_targ_disp (

   Pint          ws_type,          /* workstation type                */
   Pstore        store,            /* handle to Store object          */
   Pint          *err_ind,         /* OUT error indicator             */
   Ptarg_op_list **targ_ops        /* OUT list of target operations   */

);
```

NOTE — The memory referenced by `*targ_ops` is managed by `store`.

INQUIRE TRAVERSAL RESOURCE FACILITIES

```
void pinq_trav_res_facs (

   Pint          ws_type,          /* workstation type                */
   Pstore        store,            /* handle to Store object          */
   Pint          *err_ind,         /* OUT error indicator             */
   Ptrav_facs    **facs            /* OUT traversal resource facilities */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE CONDITIONAL TRAVERSAL FACILITIES

```
void pinq_cond_trav_facs (

   Pint       ws_type,              /* workstation type                  */
   Pstore     store,                /* handle to Store object            */
   Pint       *err_ind,             /* OUT error indicator               */
   Pcond_trav_facs  **facs,         /* OUT conditional traversal facilities
                                       */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE DIRECT INTERPRETATION TRAVERSAL FACILITIES

```
void pinq_di_trav_facs (

   Pint              ws_type,       /* workstation type                  */
   Pstore            store,         /* handle to Store object            */
   Pint              *err_ind,      /* OUT error indicator               */
   Pdi_trav_facs     **facs,        /* OUTconditionaltraversalfacilities
                                       */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE

```
void pinq_dyns_ws_attrs_texture (

   Pint                    ws_type,      /* workstation type             */
   Pint                    *err_ind,     /* OUT error indicator          */
   Pdyn_ws_attrs_texture   *dyns         /* OUT texture dynamics         */

);
```

INQUIRE DYNAMICS OF POSTING GROUPS

```
void pinq_dyns_post_grps (

   Pint                ws_type,        /* workstation type               */
   Pint                *err_ind,       /* OUT error indicator            */
   Pdyns_post_grps     *dyns           /* OUT posting dynamics           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 3

```
void pinq_def_di_pick_data3 (

   Pint              ws_type,        /* workstation type              */
   Pstore            store,          /* store for measure data        */
   Pint              *err_ind,       /* OUT error indicator           */
   Pint_list         **pet_list,     /* OUT list prompt/echo types    */
   Plimit3           *echo_vol,      /* OUT echo volume               */
   Pdi_pick_data     **pick_data     /* OUT default data record       */

);
```

NOTE — The memory referenced by `*pet_list` and `*pick_data` is managed by `store`.

INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA

```
void pinq_def_di_pick_data (

   Pint              ws_type,        /* workstation type              */
   Pstore            store,          /* store for measure data        */
   Pint              *err_ind,       /* OUT error indicator           */
   Pint_list         **pet_list,     /* OUT list prompt/echo types    */
   Plimit            *echo_area,     /* OUT echo area                 */
   Pdi_pick_data     **pick_data     /* OUT default data record       */

);
```

NOTE — The memory referenced by `*pet_list` and `*pick_data` is managed by `store`.

INQUIRE POSTING GROUP FACILITIES

```
void pinq_post_grp_facs (

   Pint              ws_type,        /* workstation type              */
   Pstore            store,          /* handle to store object        */
   Pint              *err_ind,       /* OUT error indicator           */
   Ppost_grp_facs    **facs          /* OUT posting group facilities  */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE IMAGE RESOURCE FACILITIES

```
void pinq_image_res_facs (

   Pint        ws_type,                /* workstation type              */
   Pstore      store,                  /* handle to store object        */
   Pint        *err_ind,               /* OUT error indicator           */
   Pint_list   **facs,                 /* OUT image resource facilities*/

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE LINETYPE DEFINITION SUPPORT

```
void pinq_linetype_def_support (

   Pint                ws_type,           /* workstation type          */
   Pfloat              repeat_length,     /* repeat length             */
   const Pfloat_list   *segment_lengths,  /* segment lengths           */
   Pint                *err_ind,          /* error indicator           */
   Psupport_indication*supported          /* support indication        */

);
```

INQUIRE EXTENDED PATTERN FACILITIES

```
void pinq_ext_pat_facs (

   Pint             ws_type,           /* workstation type              */
   Pstore           store,             /* handle to store object        */
   Pint             *err_ind,          /* OUT error indicator           */
   Pext_pat_facs    **facs,            /* OUT extended pattern facilities
                                          */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE HIGHLIGHTING FACILITIES

```
void pinq_highl_facs (

   Pint         ws_type,              /* workstation type                */
   Pstore          store,                /* handle to Store object       */
   Pint         *err_ind,             /* OUT error indicator             */
   Phighl_facs  *facs,                /* OUT highlighting facilities     */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE ALPHA FACILITIES

```
void pinq_alpha_facs (

   Pint           ws_type,            /* workstation type                */
   Pstore         store,              /* handle to Store object          */
   Pint           *err_ind,           /* OUT error indicator             */
   Palpha_facs    **facs              /* OUT alpha facilities            */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE TEXTURE MAPPING FACILITIES

```
void pinq_texture_map_facs (

   Pint                ws_type,           /* workstation type            */
   Pstore              store,             /* handle to Store object      */
   Pint                *err_ind,          /* OUT error indicator         */
   Ptexture_map_facs   **texture_map_facs /* OUT texture facilities      */

);
```

NOTE — The memory referenced by `*texture_map_facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TEXTURE FACILITIES

```
void pinq_texture_facs (

   Pint              ws_type,          /* workstation type              */
   Pstore            store,            /* handle to Store object        */
   Pint              *err_ind,         /* OUT error indicator           */
   Ptexture_facs     **texture_facs    /* OUT texture facilities        */

);
```

NOTE — The memory referenced by `*texture_facs` is managed by `store`.

INQUIRE MIPMAP FACILITIES

```
void pinq_mipmap_facs (

   Pint           ws_type,          /* workstation type              */
   Pint           *err_ind,         /* OUT error indicator           */
   Pmipmap_facs   *facs             /* OUT mipmap facilities         */

);
```

INQUIRE NUMBER OF PREDEFINED APPLICATION FILTERS

```
void pinq_num_pred_appl_filters (

   Pint  *err_ind,              /* OUT error indicator                 */
   Pint  *num_predefined,       /* OUT number predefined appl. filters */

);
```

INQUIRE WORKSTATION STATE TABLE LENGTHS TEXTURE

```
void pinq_ws_st_table_texture (

   Pint                   ws_type,    /* workstation type              */
   Pint                   *err_ind,   /* OUT error indicator           */
   Pws_st_tables_texture  *lengths    /* OUT lengths of workstation    */
                                      /* tables                        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE WORKSTATION STATE TABLE LENGTHS HIGHLIGHTING

```
void pinq_ws_st_table_highl (

   Pint                    ws_type,    /* workstation type              */
   Pint                    *err_ind,   /* OUT error indicator           */
   Pws_st_tables_highl     *lengths    /* OUT lengths of workstation    */
                                       /* tables                        */

);
```

INQUIRE PICK MAPPING FACILITIES

```
void pinq_pick_mapping_facs (

   Pint                 ws_type,     /* workstation type                */
   Pstore               store,       /* store for pick mapping facs     */
   Pint                 *err_ind,    /* OUT error indicator             */
   Ppick_mapping_facs   *facs        /* OUT pick mapping facilities     */

);
```

INQUIRE NUMBER OF AVAILABLE NON-ATOMIC LOGICAL INPUT DEVICES

```
void pinq_num_avail_na_in (

   Pint       ws_type,          /* workstation type                     */
   Pint       *err_ind,         /* OUT error indicator                  */
   Pnum_na_in *numin            /* OUT number of non-atomic LIDs        */

);
```

INQUIRE LOCATOR FACILITIES

```
void pinq_loc_facs (

   Pint              ws_type,     /* workstation type                   */
   Pint              device,      /* locator device number              */
   Pstore            store,       /* store                              */
   Patomic_lid_facs  **facs       /* OUT locator facilities             */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE STROKE FACILITIES

```
void pinq_stroke_facs (

   Pint              ws_type,     /* workstation type                */
   Pint              device,      /* stroke device number            */
   Pstore            store,       /* store                           */
   Pint              *err_ind,    /* OUT error indicator             */
   Patomic_lid_facs  **facs       /* OUT stroke facilities           */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE VALUATOR FACILITIES

```
void pinq_val_facs (

   Pint              ws_type,     /* workstation type                */
   Pint              device,      /* valuator device number          */
   Pstore            store,       /* store                           */
   Pint              *err_ind,    /* OUT error indicator             */
   Patomic_lid_facs  **facs       /* OUT valuator facilities         */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE CHOICE FACILITIES

```
void pinq_choice_facs (

   Pint              ws_type,     /* workstation type                */
   Pint              device,      /* choice device number            */
   Pstore            store,       /* store                           */
   Pint              *err_ind,    /* OUT error indicator             */
   Patomic_lid_facs  **facs       /* OUT choice facilities           */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PICK FACILITIES

```
void pinq_pick_facs (

   Pint              ws_type,     /* workstation type                */
   Pint              device,      /* pick device number              */
   Pstore            store,       /* store                           */
   Pint              *err_ind,    /* OUT error indicator             */
   Patomic_lid_facs  **facs       /* OUT pick facilities             */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE STRING FACILITIES

```
void pinq_string_facs (

   Pint              ws_type,     /* workstation type                */
   Pint              device,      /* string device number            */
   Pstore            store,       /* store                           */
   Pint              *err_ind,    /* OUT error indicator             */
   Patomic_lid_facs  **facs       /* OUT string facilities*/

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE SET FACILITIES

```
void pinq_set_facs (

   Pint        ws_type,           /* workstation type                */
   Pstore      store,             /* store                           */
   Pint        *err_ind,          /* OUT error indicator             */
   Pset_facs   **facs             /* OUT set facilities              */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE COMPOSITE FACILITIES

```
void pinq_composite_facs (

   Pint              ws_type,          /* workstation type              */
   Pint              device,           /* composite device number       */
   Pstore            store,            /* store                         */
   Pint              *err_ind,         /* OUT error indicator           */
   Pcomposite_facs   **facs            /* OUT composite facs            */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE PREDEFINED POSTING GROUP

```
void pinq_pred_post_grp (

   Pint              ws_type,          /* workstation type              */
   Pint              grp_id,           /* posting group identifier      */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint              *view_ind,        /* OUT default view index        */
   Ppost_grp_status  *post_grp_status, /* OUT posting group status      */
   Pbackg_style      *backg_style,     /* OUT background style          */
   Pbackg_method     *backg_method,    /* OUT background method         */
   Pborder_indic     *border_indic,    /* OUT border indicator          */
   Pint              *border_ind       /* OUT border index              */

);
```

INQUIRE PREDEFINED IMAGE RESOURCE

```
void pinq_pred_image_res (

   Pint              ws_type,          /* workstation type              */
   Pint              image_res_id,     /* image resource identifier     */
   Pstore            store,            /* handle to store object        */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint_size         *dims,            /* OUT m,n dimensions            */
   Pimage_res        **image_res,      /* OUT image resource            */
   Paccess_flag      *access_flag      /* OUT access flag               */

);
```

NOTE — The memory referenced by `*image_res` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED ASSOCIATION OF TARGET WITH TRAVERSAL RESOURCES

```
void pinq_pred_assoc_targ_trav_res (

   Pint               ws_type,              /* workstation type        */
   const Ptarg_addr   *targ_addr,           /* target address          */
   Pint               num_elems_appl_list,  /* # elems in appl. list   */
   Pint               start_ind,            /* start element to return */
   Pint               *err_ind,             /* OUT error indicator     */
   Pint_list          **res_ids,            /* OUT list of associated  */
                                            /* traversal resource ids  */
   Pint               *num_elems_impl_list  /* # elems in impl. list   */

);
```

INQUIRE PREDEFINED ASSOCIATION OF TRAVERSAL RESOURCE WITH TARGET

```
void pinq_pred_assoc_trav_res_targ (

   Pint            ws_type,          /* workstation type              */
   Pint            res_id,           /* traversal resource identifier */
   Pint            *err_ind,         /* OUT error indicator           */
   Passoc_flag     *assoc_flag,      /* OUT association flag          */
   Pint            *offset           /* OUT offset from base target   */

);
```

INQUIRE MARKER TYPE DEFINITION SUPPORT

```
void pinq_marker_type_def_support (

   Pint                ws_type,         /* workstation type          */
   const Pmarker_desc  *marker,         /* marker descriptor         */
   Pint                *err_ind,        /* error indicator           */
   Psupport_indication *supported       /* support indication        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PREDEFINED POLYLINE REPRESENTATION FULL

```
void pinq_pred_line_rep_full (

   Pint                ws_type,     /* workstation type                 */
   Pint                index,       /* predefined polyline bundle index*/
   Pint                *err_ind,    /* OUT error indicator              */
   Pline_bundle_full   *rep         /* polyline representation full     */

);
```

INQUIRE PREDEFINED EDGE REPRESENTATION FULL

```
void pinq_pred_edge_rep_full (

   Pint                ws_type,         /* workstation type             */
   Pint                index,           /* predefined edge bundle index*/
   Pint                *err_ind,        /* OUT error indicator          */
   Pedge_bundle_full   *rep             /* edge representation full     */

);
```

INQUIRE PREDEFINED EXTENDED PATTERN REPRESENTATION

```
void pinq_pred_ext_pat_rep (

   Pint                ws_type,         /* workstation type             */
   Pint                index,           /* predefined pattern index     */
   Pstore              store,           /* handle to store object       */
   Pint                *err_ind,        /* OUT error indicator          */
   Ppattern            **pattern        /* OUT pattern data             */

);
```

NOTE — The memory referenced by `*pattern` is managed by `store`..

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED HIGHLIGHTING REPRESENTATION

```
void pinq_pred_highl_rep (

   Pint              ws_type,        /* workstation type              */
   Pint              index,          /* predefined highl. index       */
   Pstore            store,          /* handle to store object        */
   Pint              *err_ind,       /* OUT error indicator           */
   Phighl_method     **method        /* OUT highl method data         */

);
```

NOTE — The memory referenced by `*method` is managed by `store`.

INQUIRE PREDEFINED TEXTURE REPRESENTATION

```
void pinq_pred_texture_rep (

   Pint              ws_type,        /* workstation type              */
   Pint              index,          /* texture index                 */
   Pstore            store,          /* handle to Store object        */
   Pint              *err_ind,       /* OUT error indicator           */
   Ptexture_rep      **rep           /* OUT texture representation    */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE PREDEFINED APPLICATION FILTER

```
void pinq_pred_appl_filter (

   Pint              ws_type,        /* workstation identifier        */
   Pint              filter_sel,     /* filter selection              */
   Pstore            store,          /* handle to store object        */
   Pint              *err_ind,       /* OUT error indicator           */
   Pfilter           **appl_filter   /* OUT application filter        */

);
```

NOTE — The memory referenced by `*appl_filter` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED SET MEASURE DEFINITION

```
void pinq_pred_set_meas_def (

   Pint             ws_type,            /* workstation type          */
   Pint             meas_proc_id,       /* set measure proc id       */
   Pint             *err_ind,           /* OUT error indicator       */
   Pint             *max_measures,      /* OUT max measures in set   */
   Pmeas_process    *meas_proc          /* OUT base measure process  */

);
```

INQUIRE PREDEFINED COMPOSITE MEASURE DEFINITION

```
void pinq_pred_composite_meas_def (

   Pint                 ws_type,        /* workstation type          */
   Pint                 meas_proc_id,   /* composite measure proc id */
   Pstore               store,          /* store for measure data    */
   Pint                 *err_ind,       /* OUT error indicator       */
   Pmeas_process_list   **meas_procs    /* OUT measure processes     */

);
```

NOTE — The memory referenced by `*meas_procs` is managed by `store`.

INQUIRE DEFAULT SET DEVICE DATA 3

```
void pinq_def_set_data3 (

   Pint             ws_type,            /* workstation type          */
   Pint             device,             /* set device number         */
   Pstore           store,              /* store for measure data    */
   Pint             *err_ind,           /* OUT error indicator       */
   Pint_list        **pet_list,         /* OUT list prompt/echo types */
   Plimit3          *echo_vol,          /* OUT echo volume           */
   Pset_data        **set_data          /* OUT default data record   */

);
```

NOTE — The memory referenced by `*pet_list` and `*set_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DEFAULT SET DEVICE DATA

```
void pinq_def_set_data (

   Pint              ws_type,          /* workstation type              */
   Pint              device,           /* set device number             */
   Pstore            store,            /* store for measure data        */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint_list         **pet_list,       /* OUT list prompt/echo types    */
   Plimit            *echo_area,       /* OUT echo area                 */
   Pset_data         **set_data        /* OUT default data record       */

);
```

NOTE — The memory referenced by `*pet_list` and `*set_data` is managed by `store`.

INQUIRE DEFAULT COMPOSITE DEVICE DATA 3

```
void pinq_def_composite_data3 (

   Pint              ws_type,          /* workstation type              */
   Pint              device,           /* composite device number       */
   Pstore            store,            /* store for measure data        */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint_list         **pet_list,       /* OUT list prompt/echo types    */
   Plimit3           *echo_vol,        /* OUT echo volume               */
   Pcomposite_data   **composite_data  /* OUT default data record       */

);
```

NOTE — The memory referenced by `*pet_list` and `*composite_data` is managed by `store`.

INQUIRE DEFAULT COMPOSITE DEVICE DATA

```
void pinq_def_composite_data (

   Pint              ws_type,          /* workstation type              */
   Pint              device,           /* composite device number       */
   Pstore            store,            /* store for measure data        */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint_list         **pet_list,       /* OUT list prompt/echo types    */
   Plimit            *echo_area,       /* OUT echo area                 */
   Pcomposite_data   **composite_data  /* OUT default data record       */

);
```

NOTE — The memory referenced by `*pet_list` and `*composite_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

13.8.5 Inquiry functions for structure state list

INQUIRE SET OF GROUPS TO WHICH POSTED

```
void pinq_grps_posted (

   Pint         ws_id,                  /* workstation identifier      */
   Pint         struct_id,              /* structure identifier        */
   Pint         num_elems_appl_list,    /* # elems in appl list        */
   Pint         start_ind,              /* starting index              */
   Pint         *err_ind,               /* OUT error indicator         */
   Pint_list    *grps,                  /* OUT list of posting groups  */
   Pint         *num_elems_impl_list    /* OUT # elems in impl list    */

);"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex A Data types in compilation order and external functions

Pages 155 to 161

A.1 Macro definitions

The following definition replaces the corresponding definition:

"

```
#define PE_NOT_STOP                          (5)   /* Ignoring function, func-
                                                      tion requires state
                                                      (PHOP,*,STOP | DISO,*) */"
```

The following definitions are added numerically by error number within the specified error macro definitions categories:

```
"/* State Errors */
#define PE_NOT_DISO                          (8)   /* Ignoring function, func-
                                                      tion requires state
                                                      (PHOP,*,DISO,*) */
#define PE_NOT_RETAINED_STOP_ONLY            (9)   /* Ignoring function, func-
                                                      tion requires state
                                                      (PHOP,*,STOP,*) */
#define PE_NOT_WSOP_AND_DISO                 (10)  /* Ignoring function, func-
                                                      tion requires state
                                                      (PHOP,WSOP,DISO,*) */

/* Workstation Errors */
#define PE_BAD_TARG_OP                       (65)  /* Ignoring function, one of
                                                      the target operations is
                                                      not supported */
#define PE_INVALID_TRAV_RES                  (66)  /* Ignoring function, the
                                                      resource is not valid */
#define PE_TRAV_RES_ASSOC_TARG               (67)  /* Ignoring function, the
                                                      traversal resource is
                                                      already associated with a
                                                      target */
#define PE_TRAV_RES_NO_DISASSOC              (68)  /* Ignoring function, the
                                                      resource cannot be disas-
                                                      sociated */
#define PE_INCOMPAT_TRAV_RES                 (69)  /* Ignoring function, source
                                                      and destination traversal
                                                      resources are incompatible
                                                      */
#define PE_DI_STRUCT_NOT_POSTED              (71)  /* Ignoring function, the
                                                      direct interpretation
                                                      structure is not posted on
                                                      the specified workstation
                                                      */
#define PE_DI_REND_TRAV_PROC_ACTIVE          (72)  /* Ignoring function, the
                                                      direct interpretation ren-
                                                      der traversal process is
                                                      already active on the
                                                      specified workstation */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_DI_REND_TRAV_PROC_NOT_ACTIVE
                                        (73)  /* Ignoring function, the
                                                 direct interpretation tra-
                                                 versal process is not
                                                 active on the specified
                                                 workstation */
#define PE_CANNOT_CREATE_TARGET         (74)  /* Ignoring function, an
                                                 additional target could
                                                 not be created */
#define PE_NUM_TARGETS_NOT_REDUCED      (75)  /* Ignoring function, the
                                                 number of targets cannot
                                                 be reduced below the
                                                 default number available
                                                 for the respective work-
                                                 station type */
#define PE_TARG_NOT_DESTROYED           (76)  /* Ignoring function, the
                                                 target could not be
                                                 destroyed */
#define PE_DISP_REND_TARG_NOT_DESTROYED (77)  /* Ignoring function, nei-
                                                 ther the display target
                                                 nor the rendering target
                                                 can be destroyed */
#define PE_MULTIPASS_IN_PROGRESS        (78)  /* Ignoring function, a
                                                 multi-pass operation is
                                                 already in progress */
#define PE_MULTI_PASS_NOT_IN_PROGRESS   (79)  /* Ignoring function, a
                                                 multi-pass operation is
                                                 not in progress */
#define PE_PASS_DEF_IN_PROGRESS         (80)  /* Ignoring function, a pass
                                                 definition is already in
                                                 progress */
#define PE_PASS_DEF_NOT_IN_PROGRESS     (81)  /* Ignoring function, a pass
                                                 definition is not in
                                                 progress */
#define PE_LT_SRC_WS_NO_OUT_CAPABILITY  (82)  /* Ignoring function, the
                                                 specified light source
                                                 workstation does not have
                                                 output capability (i.e,
                                                 the workstation category
                                                 is neither OUTPUT, OUTIN,
                                                 nor MO) */
#define PE_POST_GRP_DEF                 (83)  /* Ignoring function, the
                                                 posting group has already
                                                 been defined */
#define PE_POST_GRP_NOT_DEF             (84)  /* Ignoring function, the
                                                 posting group has not been
                                                 defined */
#define PE_POST_GRP_ID_LESS_THAN_ONE    (85)  /* Ignoring function, the
                                                 posting group identifier
                                                 is less than one */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_MAX_POST_GRP_EXCEEDED          (86)  /* Ignoring function, defining this posting group would exceed the maximum number of posting groups supported on the workstation */
#define PE_IMAGE_RES_NOT_DEF              (87)  /* Ignoring function, image resource not defined */
#define PE_BAD_IMAGE_SPEC_METHOD          (88)  /* Ignoring function, the image resource image specification method is not supported for this operation */
#define PE_IMAGE_RES_CAP_NOT_SUP          (89)  /* Ignoring function, image resource capability is not supported on the workstation */
#define PE_SPEC_IMAGE_RES_NOT_DEF         (90)  /* Ignoring function, the specified image resource has not been defined */
#define PE_MAX_IMAGE_RES_EXCEEDED         (91)  /* Ignoring function, associating this image resource would exceed the maximum number of entries allowed in the workstation image resource table */
#define PE_IMAGE_RES_EXCEEDS_CAP_WS       (92)  /* Ignoring function, the image resource exceeds the capabilities of the specified workstation */
#define PE_VW_WS_OF_CATEGORY_MI           (93)  /* Ignoring function, the specified view workstation is of category MI */
#define PE_LID_WS_NOT_IN_NOR_OUTIN        (94)  /* Ignoring function, the specified LID workstation is neither of category INPUT nor of category OUTIN */
#define PE_NEW_WS_TYPE_NOT_REALIZED       (95)  /* Warning, Not all attributes were fully realized during the creation of the new workstation type */
#define PE_WS_TYPE_CANNOT_BE_MODIFIED     (96)  /* Ignoring function, workstation type is a default type or bound to a workstation and cannot be modified */

/* Output Attribute Errors */
#define PE_NO_STATE_SAVED                 (141) /* Ignoring function, no state has been saved */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_SPEC_PAT_TYPE_NOT_SUP        (142) /* Ignoring function, the
                                                 specified pattern type is
                                                 not supported on the spec-
                                                 ified workstation */
#define PE_PAT_TYPE_NOT_SUP_ON_INQ      (143  /* Ignoring function, pat-
                                                 tern type of requested
                                                 pattern table entry is not
                                                 supported by this inquiry
                                                 */
#define PE_MAX_VIEWS_EXCEEDED           (150) /* Ignoring function, set-
                                                 ting this view table entry
                                                 would exceed the maximum
                                                 number of entries allowed
                                                 */
#define PE_INVALID_LIMITS               (151) /* Ignoring function, XMIN ≥
                                                 XMAX, YMIN ≥ YMAX, ZMIN >
                                                 ZMAX, UMIN ≥ UMAX or VMIN ≥
                                                 VMAX */
#define PE_INVALID_VIEWPORT             (152) /* Ignoring function, invalid
                                                 viewport: XMIN ≥ XMAX, YMIN
                                                 ≥ YMAX, ZMIN > ZMAX */
#define PE_INVALID_VIEW_CLIP_LIM        (153) /* Ignoring function, invalid
                                                 view clipping limits: XMIN
                                                 ≥ XMAX, YMIN ≥ YMAX, ZMIN >
                                                 ZMAX */
#define PE_VIEW_CLIP_LIM_NOT_NPC        (154) /* Ignoring function, the
                                                 view clipping limits are
                                                 not within NPC range */
#define PE_PROJ_VP_LIMITS_NOT_NPC       (155) /* Ignoring function, the
                                                 projection viewport lim-
                                                 its are not within NPC
                                                 range */
#define PE_WS_WIN_LIM_NOT_NPC           (156) /* Ignoring function, the
                                                 workstation window limits
                                                 are not within NPC range
                                                 */
#define PE_WS_VP_NOT_IN_DISPLAY_SPACE   (157) /* Ignoring function, the
                                                 workstation viewport is
                                                 not within display space
                                                 */
#define PE_INVALID_VIEW_PLANES          (158) /* Ignoring function, front
                                                 plane and back plane dis-
                                                 tances same and z-extent
                                                 of projection viewport
                                                 non-zero */
#define PE_VPN_LEN_ZERO                 (159) /* Ignoring function, the
                                                 view plane normal has
                                                 length zero */
#define PE_VUP_LEN_ZERO                 (160) /* Ignoring function, the
                                                 view up vector has lenght
                                                 zero */
#define PE_VUP_VPN_PARALLEL             (161) /* Ignoring function, view up
                                                 and view plane normal vec-
                                                 tors parallel */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_PRP_BETW_FRONT_BACK              (162) /* Ignoring function, the
                                                     projection reference point
                                                     is between the front and
                                                     back planes */
#define PE_PRP_ON_VIEW_PLANE                (163) /* Ignoring function, the
                                                     projection referece point
                                                     cannot be positioned on
                                                     the view plane */
#define PE_BACK_PLN_BEFORE_FRONT_PLN        (164) /* Ignoring function, the
                                                     back plane is in front of
                                                     the front plane */
#define PE_HIGHL_IND_LT_ONE                 (165) /* Ignoring function, the
                                                     highlighting index is less
                                                     than one */
#define PE_HIGHL_METHOD_NOT_AVAIL           (166) /* Ignoring function, the
                                                     specified highlighting
                                                     method is not available on
                                                     the specified workstation
                                                     */
#define PE_LINETYPE_NOT_DEFINABLE           (167) /* Ignoring function, the
                                                     specified linetype does
                                                     not reference a definable
                                                     linetype */
#define PE_MAX_DEF_LINETYPES_EXCEEDED       (168) /* Ignoring function, set-
                                                     ting this table entry
                                                     would exceed the maximum
                                                     number of entries allowed
                                                     in the definable linetype
                                                     table */
#define PE_LINETYPE_NOT_DEFINED             (169) /* Ignoring function, the
                                                     specified linetype has not
                                                     been defined */
#define PE_LINETYPE_NOT_SUP                 (170) /* Warning, this linetype
                                                     definition is not sup-
                                                     ported on at least one of
                                                     the available workstation
                                                     types */
#define PE_LINETYPE_DEF_NOT_AVAIL           (171) /* Ignoring function, the
                                                     specified standard  or
                                                     implementation-defined
                                                     linetype definition is not
                                                     available */
#define PE_MARKER_TYPE_NOT_DEFINABLE        (172) /* Ignoring function, the
                                                     specified marker type does
                                                     not reference a definable
                                                     marker type */
#define PE_MAX_DEF_MARKER_TYPES_EXCEEDED    (173) /* Ignoring function, set-
                                                     ting this table entry
                                                     would exceed the maximum
                                                     number of entries allowed
                                                     in the definable marker
                                                     type table */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_MARKER_TYPE_NOT_DEFINED          (174) /* Ignoring function, the
                                                     specified marker type has
                                                     not been defined */
#define PE_MARKER_SHAPE_INVALID             (175) /* Ignoring function, the
                                                     marker shape descriptor is
                                                     invalid */
#define PE_MARKER_SHAPE_TYPE_NOT_SUP        (176) /* Ignoring function, the
                                                     specified marker shape
                                                     type is not supported */
#define PE_MARKER_TYPE_DEF_NOT_AVAIL        (177) /* Ignoring function, the
                                                     specified standard or
                                                     implementation defined
                                                     marker type definition is
                                                     not available */
#define PE_FILTER_SEL_INVALID               (178) /* Ignoring function, the
                                                     filter selection is
                                                     invalid */
#define PE_FILTER_NOT_DEFINED               (179) /* Ignoring function, the
                                                     specified application fil-
                                                     ter has not been defined
                                                     */

/* Input Errors */
#define PE_DI_PICK_TRAV_PROC_ACTIVE         (264) /* Ignoring function, the
                                                     direct interpretation pick
                                                     traversal process is
                                                     already active on the
                                                     specified workstation */
#define PE_DI_PICK_TRAV_PROC_NOT_ACTIVE     (265) /* Ignoring function, the
                                                     direct interpretation pick
                                                     traversal process is not
                                                     active on the specified
                                                     workstation */
#define PE_BAD_PICK_TYPE                    (266) /* Ignoring function, the
                                                     specified pick type is not
                                                     available on the speci-
                                                     fied workstation */
#define PE_LISTED_MEAS_PROC_INVALID         (267) /* Ignoring function, one of
                                                     the listed measure pro-
                                                     cesses is invalid */
#define PE_LISTED_TRIG_PROC_INVALID         (268) /* Ignoring function, one of
                                                     the listed trigger pro-
                                                     cesses is invalid */
#define PE_LISTED_ECHO_PROC_INVALID         (269) /* Ignoring function, one of
                                                     the listed echo processes
                                                     is invalid */
#define PE_LISTED_ACK_PROC_INVALID          (270) /* Ignoring function, one of
                                                     the listed acknowledge-
                                                     ment processes is invalid
                                                     */
#define PE_LID_NR_IN_USE                    (271) /* Ignoring function, device
                                                     number already in use */
#define PE_NR_MEAS_INVALID                  (272) /* Ignoring function, number
                                                     of measures is invalid */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_MEAS_ID_NOT_AVAIL                (273) /* Ignoring function, the
                                                     specified measure identi-
                                                     fier is not available on
                                                     the specified workstation
                                                     */
#define PE_MEAS_ID_IN_USE                   (274) /* Ignoring function, the
                                                     specified measure identi-
                                                     fier is currently part of
                                                     a set or composite logi-
                                                     cal input device defini-
                                                     tion */
#define PE_NO_PICK_MEASURE                  (275) /* Ignoring function, the
                                                     specified set or compos-
                                                     ite device does not have a
                                                     pick measure */
#define PE_LID_INAPPROP_FOR_VW_ACTION       (276) /* Ignoring function, the
                                                     logical input device class
                                                     specified is inappropri-
                                                     ate for the view action
                                                     specified */
#define PE_LID_INAPPROP_FOR_LT_ACT          (277) /* Ignoring function, the
                                                     input device class speci-
                                                     fied is inappropriate for
                                                     the light action speci-
                                                     fied */
#define PE_LT_ACT_INAPPROP_FOR_LT_SRC       (278) /* Ignoring function, the
                                                     light action is inappro-
                                                     priate for the light
                                                     source specified */

/* Miscellaneous Errors */
#define PE_TEST_OP_INVALID                  (451) /* Ignoring function, invalid
                                                     test operator for test type
                                                     */
#define PE_NO_WINDOW_SYSTEM                 (452) /* Ignoring function, the
                                                     specified workstation is
                                                     not operating in a window
                                                     management system envrion-
                                                     ment */
#define PE_WIN_SYS_COLR_NOT_DETER           (453) /* Ignoring function, the
                                                     window system colour can-
                                                     not be determined */
#define PE_TEXTR_COORD_SRC_NOT_AVAIL        (600) /* Ignoring function, the
                                                     specified coordinate
                                                     source is not available on
                                                     the specified workstation
                                                     */
#define PE_DATA_VALUE_IND_LESS_1            (601) /* Ignoring function, the
                                                     specified data value index
                                                     is less than one */
#define PE_TEXTR_COMPOS_METHOD_NOT_AVAIL
                                            (602) /* Ignoring function, the
                                                     specified composition
                                                     method is not available on
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                                        the specified workstation
                                        */
#define PE_TEXTR_MIN_METHOD_NOT_AVAIL   (603) /* Ignoring function, the
                                        specified minification
                                        method is not available on
                                        the specified workstation
                                        */
#define PE_TEXTR_MAG_METHOD_NOT_AVAIL   (604) /* Ignoring function, the
                                        specified magnification
                                        method is not available on
                                        the specified workstation
                                        */
#define PE_BOUND_COND_NOT_AVAIL         (605) /* Ignoring function, the
                                        specified boundary condi-
                                        tion is not available on
                                        the specified workstation
                                        */
#define PE_BOUND_CLAMP_METHOD_NOT_AVAIL
                                        (606) /* Ignoring function, the
                                        specified boundary clamp
                                        method is not available on
                                        the specified workstation
                                        */
#define PE_DEPTH_SAMPL_BIAS_OUT_OF_RANGE
                                        (607) /* Ignoring function, the
                                        specified depth sampling
                                        bias hint is out of range
                                        */
#define PE_TEXTR_IMAGE_RES_NOT_DEFINED  (608) /* Ignoring function, the
                                        specified texture image
                                        resource identifier is not
                                        defined on the specified
                                        workstation */
#define PE_RENDERING_ORDER_NOT_AVAIL    (612) /* Ignoring function, the
                                        specified rendering order
                                        is not available on the
                                        specified workstation */"
```

Pages 161 to 164

The following definitions are added numerically by function number to the list of function name macro definitions:

```
#define Pfn_ws_type_create                  (268)
#define Pfn_ws_type_set                     (269)
#define Pfn_ws_type_get                     (270)
#define Pfn_ws_type_destroy                 (271)
#define Pfn_redraw_all_structs_from_grp     (272)
#define Pfn_redraw_all_structs_on_targ      (273)
#define
Pfn_redraw_all_structs_from_grp_on_targ     (274)
#define Pfn_upd_targ                        (275)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define Pfn_define_post_grp                (276)
#define Pfn_undefine_post_grp              (277)
#define Pfn_set_post_grp_status            (278)
#define Pfn_set_post_grp_priority          (279)
#define Pfn_set_post_grp_backg_style       (280)
#define Pfn_set_post_grp_backg_method      (281)
#define Pfn_set_post_grp_border_indic      (282)
#define Pfn_set_post_grp_border_ind        (283)
#define Pfn_assoc_image_res                (284)
#define Pfn_disassoc_image_res             (285)
#define Pfn_set_dc_clip_regions3           (286)
#define Pfn_set_dc_clip_regions            (287)
#define Pfn_set_targ_manip_mode            (288)
#define Pfn_set_targ_dispos                (289)
#define Pfn_set_disp_targ                  (290)
#define Pfn_set_rend_targ                  (291)
#define Pfn_clear_targ                     (292)
#define Pfn_copy_targ                      (293)
#define Pfn_create_targ                    (294)
#define Pfn_destroy_targ                   (295)
#define Pfn_set_st_visual_rep              (296)
#define Pfn_set_targ_st_visual_rep         (297)
#define Pfn_assoc_trav_res                 (298)
#define Pfn_disassoc_trav_res              (299)
#define Pfn_manip_trav_res                 (300)
#define Pfn_reset_all_trav_res             (301)
#define Pfn_ret_window_system_colr         (302)
#define Pfn_set_appl_int                   (303)
#define Pfn_set_appl_real                  (304)
#define Pfn_circle3                        (305)
#define Pfn_circle                         (306)
#define Pfn_circular_arc3                  (307)
#define Pfn_circular_arc                   (308)
#define Pfn_ellipse3                       (309)
#define Pfn_ellipse                        (310)
#define Pfn_elliptical_arc3                (311)
#define Pfn_elliptical_arc                 (312)
#define Pfn_fill_circle3                   (313)
#define Pfn_fill_circle                    (314)
#define Pfn_circular_arc_close3            (315)
#define Pfn_circular_arc_close             (316)
#define Pfn_fill_ellipse3                  (317)
#define Pfn_fill_ellipse                   (318)
#define Pfn_elliptical_arc_close3          (319)
#define Pfn_elliptical_arc_close           (320)
#define Pfn_set_highl_ind                  (321)
#define Pfn_set_linetype_adapt             (322)
#define Pfn_set_linetype_cont              (323)
#define Pfn_set_linetype_offset            (324)
#define Pfn_set_linecap                    (325)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_set_linejoin                    (326)
#define Pfn_set_linemitre_limit             (327)
#define Pfn_set_edgetype_adapt              (328)
#define Pfn_set_edgetype_cont               (329)
#define Pfn_set_edgetype_offset             (330)
#define Pfn_set_edgecap                     (331)
#define Pfn_set_edgejoin                    (332)
#define Pfn_set_edgemitre_limit             (333)
#define Pfn_set_highl_method                (334)
#define Pfn_set_transparency                (335)
#define Pfn_set_back_transparency           (336)
#define Pfn_set_alpha_src_sel               (337)
#define Pfn_set_alpha_data_sel_ind          (338)
#define Pfn_set_active_textures             (339)
#define Pfn_set_back_active_textures        (340)
#define Pfn_set_texture_perspect_corr       (341)
#define Pfn_set_texture_sampling_freq       (342)
#define Pfn_set_texture_res_opt_heur        (343)
#define Pfn_set_cond_flags                  (344)
#define Pfn_set_cond_flags_from_tests       (345)
#define Pfn_set_line_rep_full               (346)
#define Pfn_set_line_rep_mask               (347)
#define Pfn_set_marker_rep_mask             (348)
#define Pfn_set_text_rep_mask               (349)
#define Pfn_set_int_rep_mask                (350)
#define Pfn_set_edge_rep_full               (351)
#define Pfn_set_edge_rep_mask               (352)
#define Pfn_set_ext_pat_rep                 (353)
#define Pfn_set_pat_rep_mask                (354)
#define Pfn_set_texture_rep                 (355)
#define Pfn_set_texture_rep_mask            (356)
#define Pfn_set_texture_param               (357)
#define Pfn_set_texture_composition         (358)
#define Pfn_set_texture_sampling            (359)
#define Pfn_set_texture_binding             (360)
#define Pfn_set_highl_rep                   (361)
#define Pfn_set_transparency_mode           (362)
#define Pfn_set_transparency_thresholds     (363)
#define Pfn_set_refl_rep_mask               (364)
#define Pfn_set_appl_filter                 (365)
#define Pfn_create_mipmap_texture           (366)
#define Pfn_define_linetype                 (367)
#define Pfn_define_marker_type              (368)
#define Pfn_set_view_ref_point3             (369)
#define Pfn_set_view_ref_point              (370)
#define Pfn_set_view_plane_norm             (371)
#define Pfn_set_view_up_vec3                (372)
#define Pfn_set_view_up_vec                 (373)
#define Pfn_set_view_win_limits             (374)
#define Pfn_set_proj_vp3                    (375)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_set_proj_vp                          (376)
#define Pfn_set_proj_type                        (377)
#define Pfn_set_proj_ref_point                   (378)
#define Pfn_set_view_plane_dist                  (379)
#define Pfn_set_front_plane_dist                 (380)
#define Pfn_set_back_plane_dist                  (381)
#define Pfn_set_xy_clip_indicator                (382)
#define Pfn_set_front_clip_indicator             (383)
#define Pfn_set_back_clip_indicator              (384)
#define Pfn_upd_view_rep                         (385)
#define Pfn_map_dc_to_wsc                        (386)
#define Pfn_map_wsc_to_dc                        (387)
#define Pfn_map_dc_to_wc                         (388)
#define Pfn_open_di_struct                       (389)
#define Pfn_close_di_struct                      (390)
#define Pfn_inst_struct                          (391)
#define Pfn_cond_exec_struct                     (392)
#define Pfn_cond_inst_struct                     (393)
#define Pfn_cond_return                          (394)
#define Pfn_cond_skip_elements                   (395)
#define Pfn_cond_skip_to_label                   (396)
#define Pfn_push_st                              (397)
#define Pfn_pop_st                               (398)
#define Pfn_copy_elem_struct                     (399)
#define Pfn_copy_elem_range_struct               (400)
#define Pfn_copy_elems_between_labels_struct(401)
#define Pfn_move_elem                            (402)
#define Pfn_move_elem_range                      (403)
#define Pfn_move_elems_between_labels            (404)
#define Pfn_set_watch_on_elem_range              (405)
#define Pfn_end_watch_on_elem_range              (406)
#define Pfn_post_struct_to_grp                   (407)
#define Pfn_unpost_structs_from_grps             (408)
#define Pfn_unpost_all_structs_from_grp          (409)
#define Pfn_post_di_struct                       (410)
#define Pfn_post_di_struct_to_grp                (411)
#define Pfn_unpost_di_struct                     (412)
#define Pfn_renew_di_state                       (413)
#define Pfn_set_di_mode                          (414)
#define Pfn_mark_multi_pass_start                (415)
#define Pfn_mark_multi_pass_compl                (416)
#define Pfn_mark_pass_start                      (417)
#define Pfn_mark_pass_compl                      (418)
#define Pfn_ret_num_passes_req                   (419)
#define Pfn_set_set_pick_filter                  (420)
#define Pfn_set_composite_pick_filter            (421)
#define Pfn_set_di_pick_filter                   (422)
#define Pfn_set_di_pick_corr_point3              (423)
#define Pfn_set_di_pick_corr_point               (424)
#define Pfn_set_pick_mapping_data                (425)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_map_dc_point_to_pick_paths      (426)
#define Pfn_define_locator                  (427)
#define Pfn_define_stroke                   (428)
#define Pfn_define_valuator                 (429)
#define Pfn_define_choice                   (430)
#define Pfn_define_pick                     (431)
#define Pfn_define_string                   (432)
#define Pfn_create_set_measure              (433)
#define Pfn_define_set                      (434)
#define Pfn_create_composite_measure        (435)
#define Pfn_define_composite                (436)
#define Pfn_undefine_locator                (437)
#define Pfn_undefine_stroke                 (438)
#define Pfn_undefine_valuator               (439)
#define Pfn_undefine_choice                 (440)
#define Pfn_undefine_pick                   (441)
#define Pfn_undefine_string                 (442)
#define Pfn_destroy_set_measure             (443)
#define Pfn_undefine_set                    (444)
#define Pfn_destroy_composite_measure       (445)
#define Pfn_undefine_composite              (446)
#define Pfn_init_set3                       (447)
#define Pfn_init_set                        (448)
#define Pfn_init_composite3                 (449)
#define Pfn_init_composite                  (450)
#define Pfn_set_set_mode                    (451)
#define Pfn_set_composite_mode              (452)
#define Pfn_req_set3                        (453)
#define Pfn_req_set                         (454)
#define Pfn_req_composite3                  (455)
#define Pfn_req_composite                   (456)
#define Pfn_sample_set3                     (457)
#define Pfn_sample_set                      (458)
#define Pfn_sample_composite3               (459)
#define Pfn_sample_composite                (460)
#define Pfn_get_set3                        (461)
#define Pfn_get_set                         (462)
#define Pfn_get_composite3                  (463)
#define Pfn_get_composite                   (464)
#define Pfn_attach_lid_to_view              (465)
#define Pfn_detach_lid_from_view            (466)
#define Pfn_attach_lid_to_light_src         (467)
#define Pfn_detach_lid_from_light_src       (468)
#define Pfn_init_di_pick3                   (469)
#define Pfn_init_di_pick                    (470)
#define Pfn_enable_di_pick                  (471)
#define Pfn_disable_di_pick                 (472)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_trans_di_pick_set                    (473)
"
```

Page 164

The following text is appended after the definition of PLINE_DASH_DOT:

```
"
#define PLINE_DASH_DOT_DOT                       (5)
"
```

Page 165 (Page 99 of PHIGS PLUS)

The following text is appended the definition of PCOLR_HLS:

```
"
#define PCOLR_RGBA                               (5)
#define PCOLR_CIELUVA                            (6)
#define PCOLR_HSVA                               (7)
#define PCOLR_HLSA                               (8)
"
```

Page 166

The following text is appended:

```
"
#define PHATCH_HORIZ                             (1)
#define PHATCH_VERT                              (2)
#define PHATCH_POS_SLOPE                         (3)
#define PHATCH_NEG_SLOPE                         (4)
#define PHATCH_HORIZ_VERT_CROSS                  (5)
#define PHATCH_POS_NEG_SLOPE_CROSS               (6)

#define PPICK_HIGHL_PRIM                         (4)
#define PPICK_HIGHL_PRIM_GROUP                   (5)
#define PPICK_HIGHL_STRUCT_NETWORK               (6)

#define PDI_PICK_HIGHL                           (1)

#define PPICK_TYPE_WS_DEP                        (1)
#define PPICK_TYPE_PICK_FIRST                    (2)
#define PPICK_TYPE_PICK_LAST                     (3)
#define PPICK_TYPE_PICK_CLOSEST                  (4)

#define PTARG_MANIP_AUTO                         (1)
#define PTARG_MANIP_MANUAL                       (2)

#define PSET_REND_TARG                           (1)
#define PSET_DISP_TARG                           (2)
#define PCLEAR_TARG                              (3)
#define PCOPY_TARG                               (4)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PCLEAR_TRAV_RES                         (0)
#define PCOPY_TRAV_RES                          (1)

#define PTRAV_RES_Z_BUF                         (1)
#define PTRAV_RES_PAINT_BUF                     (2)

#define PVUM_APPLY_ALL                          (1)
#define PVUM_APPLY_ORIENTATION                  (2)

#define PVWA_ROT_ABOUT_U_VRC_AXIS               (1)
#define PVWA_ROT_ABOUT_V_VRC_AXIS               (2)
#define PVWA_ROT_ABOUT_N_VRC_AXIS               (3)
#define PVWA_TRANS_U_VRC_AXIS_CLIP              (4)
#define PVWA_TRANS_U_VRC_AXIS_NOCLIP            (5)
#define PVWA_TRANS_V_VRC_AXIS_CLIP              (6)
#define PVWA_TRANS_V_VRC_AXIS_NOCLIP            (7)
#define PVWA_TRANS_N_VRC_AXIS                   (8)
#define PVWA_SCALE_WIN_UNIFORMLY                (9)
#define PVWA_SCALE_VP_UNIFORMLY                 (10)
#define PVWA_SCALE_WIN_WIDTH                    (11)
#define PVWA_SCALE_VP_WIDTH                     (12)
#define PVWA_SCALE_WIN_HEIGHT                   (13)
#define PVWA_SCALE_VP_HEIGHT                    (14)
#define PVWA_TRANS_FRONT_CLIP_PLANE             (15)
#define PVWA_TRANS_BACK_CLIP_PLANE              (16)
#define PVWA_ROT_AROUND_VRP_ALONG_CYL_LAT       (17)
#define PVWA_TRANS_EYE_FROM_VRP_OUT             (18)
#define PVWA_TRANS_EYE_FROM_VRP_UP              (19)
#define PVWA_ROT_EYE_FROM_VRP_UP                (20)
#define PVWA_CHG_EYE_POS_WRT_VRP                (21)
#define PVWA_CHG_VRP_POS_ROT_UP_WRT_PRP         (22)
#define PVWA_CHG_VRP_POS_ROT_RL_WRT_PRP         (23)

#define PMTT_MEASURE_CHANGE                     (1)
#define PMTT_EVENT_WITH_DISCARD                 (2)
#define PMTT_EVENT_WITH_PROPAGATION             (3)

#define PBACKGM_COLOUR_TABLE_0                  (1)
#define PBACKGM_COLOUR_INDEX                    (2)
#define PBACKGM_COLOUR                          (3)
#define PBACKGM_IMAGE_RESOURCE                  (4)

#define PIMGSM_UNCOMP_COLR_INDEX_ARRAY          (1)
#define PIMGSM_UNCOMP_COLR_ARRAY                (2)
#define PIMGSM_WINDOW_SYS_BITMAP                (3)
#define PIMGSM_VIDEO_SIGNAL_CHAN_ID             (4)
#define PIMGSM_LUMINANCE                        (5)
#define PIMGSM_LUMINANCE_ALPHA                  (6)
#define PIMGSM_MIPMAP_COLRS                     (7)
#define PIMGSM_MIPMAP_LUMINANCE                 (8)
#define PIMGSM_MIPMAP_LUMINANCE_ALPHA           (9)

#define PPATT_COLOUR_INDEX                      (1)
#define PPATT_COLOUR                            (2)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PPATT_IMAGE_RESOURCE                    (3)

#define PSHAPE_POLYLINE                         (1)
#define PSHAPE_FILL_AREA                        (2)
#define PSHAPE_CONVEX_FILL_AREA                 (3)

#define PTEST_ALWAYS_FAIL                       (0)
#define PTEST_ALWAYS_SUCCEED                    (1)
#define PTEST_EXTENT3                           (2)
#define PTEST_EXTENT                            (3)
#define PTEST_BOUNDS3                           (4)
#define PTEST_BOUNDS                            (5)
#define PTEST_NAMESET                           (6)
#define PTEST_APPL_INTEGER                      (7)
#define PTEST_APPL_REAL                         (8)

#define PFILTER_TYPE_INVIS                      (1)
#define PFILTER_TYPE_HIGHL                      (2)
#define PFILTER_TYPE_PICK                       (3)
#define PFILTER_TYPE_APPL                       (4)

#define PTEXTRSF_PER_PIXEL                      (1)
#define PTEXTRSF_INTERP_DEP                     (2)

#define PTEXTR_OPT_NO_HINT                      (1)
#define PTEXTR_OPT_SPEED                        (2)
#define PTEXTR_OPT_SPACE                        (3)

#define PALPHASRC_TRANSPARENCY                  (1)
#define PALPHASRC_COLR_ASPECT                   (2)
#define PALPHASRC_FACET_COLR                    (3)
#define PALPHASRC_FACET_DATA                    (4)
#define PALPHASRC_VERTEX_COLR                   (5)
#define PALPHASRC_VERTEX_DATA                   (6)

#define PCOORDSRC_VERTEX_COORD_MC               (1)
#define PCOORDSRC_VERTEX_COORD_WC               (2)
#define PCOORDSRC_VERTEX_NORMAL_MC              (3)
#define PCOORDSRC_VERTEX_NORMAL_WC              (4)
#define PCOORDSRC_NURB_SURF_UV                  (5)
#define PCOORDSRC_DATA                          (6)
#define PCOORDSRC_REFL_SPHERE_VRC               (7)
#define PCOORDSRC_REFL_SPHERE_WC                (8)

#define PGCS_GAMMA_ONE                          (1)
#define PGCS_GAMMA_RED                          (1)
#define PGCS_GAMMA_TWO                          (2)
#define PGCS_GAMMA_GREEN                        (2)
#define PGCS_GAMMA_THREE                        (3)
#define PGCS_GAMMA_BLUE                         (3)

#define PCOMPOS_REPLACE                         (1)
#define PCOMPOS_MODULATE                        (2)
#define PCOMPOS_BLEND_ENVIRONMENT_COLR          (3)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PCOMPOS_DECAL                          (4)
#define PCOMPOS_DECAL_BACKGROUND               (5)

#define PMINM_SINGLE_BASE                      (1)
#define PMINM_LINEAR_BASE                      (2)
#define PMINM_SINGLE_IN_MIPMAP                 (3)
#define PMINM_LINEAR_IN_MIPMAP                 (4)
#define PMINM_SINGLE_BETWEEN_MIPMAPS           (5)
#define PMINM_LINEAR_BETWEEN_MIPMAPS           (6)

#define PMAGM_SINGLE_BASE                      (1)
#define PMAGM_LINEAR_BASE                      (2)
#define PMAGM_SINGLE_IN_MIPMAP                 (3)
#define PMAGM_LINEAR_IN_MIPMAP                 (4)
#define PMAGM_SINGLE_BETWEEN_MIPMAPS           (5)
#define PMAGM_LINEAR_BETWEEN_MIPMAPS           (6)

#define PBOUND_COND_CLAMP_EXPLICIT_METHOD      (1)
#define PBOUND_COND_CLAMP_BORDER               (2)
#define PBOUND_COND_WRAP                       (3)
#define PBOUND_COND_MIRROR                     (4)

#define PBOUND_CLAMP_DISCONTINUE               (1)
#define PBOUND_CLAMP_COLR                      (2)

#define PRENDORDR_PRELIGHT                     (1)
#define PRENDORDR_POST_LIGHT                   (2)

#define PTRANSPMODE_NONE                       (1)
#define PTRANSPMODE_SCREEN_DOOR                (2)
#define PTRANSPMODE_ALPHA_BLEND                (3)
#define PTRANSPMODE_TWO_PASS                   (4)
#define PTRANSPMODE_MULTIPASS                  (5)

#define PLTA_ADJ_LT_COLR_COMP_1                (1)
#define PLTA_ADJ_LT_COLR_COMP_2                (2)
#define PLTA_ADJ_LT_COLR_COMP_3                (3)
#define PLTA_ADJ_BRIGHT                        (4)
#define PLTA_ROT_AROUND_X_DIR_VEC              (5)
#define PLTA_ROT_AROUND_Y_DIR_VEC              (6)
#define PLTA_ROT_AROUND_Z_DIR_VEC              (7)
#define PLTA_TRANS_X_LT_POS                    (8)
#define PLTA_TRANS_Y_LT_POS                    (9)
#define PLTA_TRANS_Z_LT_POS                    (10)
#define PLTA_ADJ_CONCENTRATION_EXP             (11)
#define PLTA_ADJ_LT_SPREAD_ANGLE               (12)"
```

Page 169

A.2 Types in compilation order

The following text replaces the definition of Pint_style*:*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

“

```
typedef enum {
  /* start of PHIGS enumerations */
  PSTYLE_HOLLOW,
  PSTYLE_SOLID,
  PSTYLE_HATCH,
  PSTYLE_PATTERN,
  PSTYLE_EMPTY,
  /* end of PHIGS enmerations */
  /* start of Full PHIGS enumerations */
  PSTYLE_TEXTURE
  /* end of Full PHIGS enumerations */
} Pint_style;”
```

The following text is merged into the definition of Paspect*:*

“

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  ...,
  /* end of PHIGS PLUS enumerations */
  /* start of Full PHIGS enumerations */
  PASPECT_HIGHL_METHOD
  /* end of Full PHIGS enumerations */
} Paspect;”
```

Page 171

The following text replaces the text of the definition of popen_struct_status*:*

“

```
typedef enum {
  /* start of PHIGS enumerations */
  PSTRUCT_NONE,
  PSTRUCT_OPEN,
  /* end of PHIGS enumerations */
  /* start of Full PHIGS enumerations */
  PSTRUCT_DI
  /* end of Full PHIGS enumerations */
} Popen_struct_status;”
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The following text replaces the text of the definition of pin_class*:*

"

```
typedef enum {
    /* start of Basic PHIGS enumerations */
    PIN_NONE,
    PIN_LOC,
    PIN_STROKE,
    PIN_VAL,
    PIN_CHOICE,
    PIN_PICK,
    PIN_STRING,
    /* end of Basic PHIGS enumerations */
    /* start of Full PHIGS enumerations */
    PIN_SET,
    PIN_COMPOSITE
    /* end of Full PHIGS enumerations */
} Pin_class;"
```

Page 174

The following text is merged into the text of the definition of pelem_type*:*

"

```
typedef enum {
    /* start of PHIGS enumerations */
    ...,
    /* end of PHIGS enumerations */
    /* start of PHIGS PLUS enumerations */
    ...,
    /* end of PHIGS PLUS enumerations */
    /* start of Full PHIGS enumerations */
    PELEM_APPL_INT,
    PELEM_APPL_REAL,
    PELEM_CIRCLE3,
    PELEM_CIRCLE,
    PELEM_CIRCULAR_ARC3,
    PELEM_CIRCULAR_ARC,
    PELEM_ELLIPSE3,
    PELEM_ELLIPSE,
    PELEM_ELLIPTICAL_ARC3,
    PELEM_ELLIPTICAL_ARC,
    PELEM_FILL_CIRCLE3,
    PELEM_FILL_CIRCLE,
    PELEM_CIRCULAR_ARC_CLOSE3,
    PELEM_CIRCULAR_ARC_CLOSE,
    PELEM_FILL_ELLIPSE3,
    PELEM_FILL_ELLIPSE,
    PELEM_ELLIPTICAL_ARC_CLOSE3,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    PELEM_ELLIPTICAL_ARC_CLOSE,
    PELEM_HIGHL_IND,
    PELEM_LINETYPE_ADAPT,
    PELEM_LINETYPE_CONT,
    PELEM_LINETYPE_OFFSET,
    PELEM_LINECAP,
    PELEM_LINEJOIN,
    PELEM_LINEMITRE_LIMIT,
    PELEM_EDGETYPE_ADAPT,
    PELEM_EDGETYPE_CONT,
    PELEM_EDGETYPE_OFFSET,
    PELEM_EDGECAP,
    PELEM_EDGEJOIN,
    PELEM_EDGEMITRE_LIMIT,
    PELEM_HIGHL_METHOD,
    PELEM_TRANSPARENCY,
    PELEM_BACK_TRANSPARENCY,
    PELEM_ALPHA_SRC_SEL,
    PELEM_ALPHA_DATA_SEL_IND,
    PELEM_ACTIVE_TEXTURES,
    PELEM_BACK_ACTIVE_TEXTURES,
    PELEM_TEXTURE_PERSPECT_CORR,
    PELEM_TEXTURE_SAMPLING_FREQ,
    PELEM_TEXTURE_RES_OPT_HEUR,
    PELEM_SET_COND_FLAGS,
    PELEM_SET_COND_FLAGS_FROM_TESTS,
    PELEM_INST_STRUCT,
    PELEM_COND_EXEC_STRUCT,
    PELEM_COND_INST_STRUCT,
    PELEM_COND_RETURN,
    PELEM_COND_SKIP_ELEMS,
    PELEM_COND_SKIP_TO_LABEL,
    PELEM_PUSH_ST,
    PELEM_POP_ST
    /* end of Full PHIGS enumerations */

} Pelem_type;"
```

Page 175

The following text is inserted after the definition of Prgb:

"

```
typedef struct {

    Pfloat      cieluv_x;       /* x coefficient                */
    Pfloat      cieluv_y;       /* y coefficient                */
    Pfloat      cieluv_y_lum;   /* y luminance                  */
    Pfloat      alpha;          /* alpha                        */

} Pcieluva;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

   Pfloat     hue;            /* hue                                  */
   Pfloat     lightness;      /* lightness                            */
   Pfloat     saturation;     /* saturation                           */
   Pfloat     alpha;          /* alpha                                */

} Phlsa;

typedef struct {

   Pfloat     hue;            /* hue                                  */
   Pfloat     saturation;     /* saturation                           */
   Pfloat     value;          /* value                                */
   Pfloat     alpha;          /* alpha                                */

} Phsva;

typedef struct {

   Pfloat     red;            /* red intensity                        */
   Pfloat     green;          /* green intensity                      */
   Pfloat     blue;           /* blue intensity                       */
   Pfloat     alpha;          /* alpha                                */

} Prgba;"
```

The following text replaces the definition of Pcolr_rep*:*

```
"
typedef union {

   /* start of Basic PHIGS union members */
   Prgb       rgb;        /* RGB colour specification                */
   Pcieluv    cieluv;     /* CIE L*u*v* colour specification         */
   Phls       hls;        /* HLS colour specification                */
   Phsv       hsv;        /* HSV colour specification                */
   /* end of Basic PHIGS union members */
   /* start of Full PHIGS union members */
   Prgba      rgba;       /* RGB specification with alpha            */
   Pcieluva   cieluva;    /* CIE L*u*v* colour specfication with alpha*/
   Phlsa      hlsa;       /* HLS colour specificaiton with alpha     */
   Phsva      hsva;       /* HSV colour specification with alpha     */
   /* end of Full PHIGS union members */
   Pdata      unsupp;     /* colour in an unsupported colour model   */
   int        impl_dep;   /* implementation defined                  */

} Pcolr_rep;"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Page 187 through 193

The following type definitions replace the corresponding type definitions:

"

```
typedef struct {
   union Pchoice_pets {
      struct Pchoice_pet_other {
         Pint      unused;
      } pet_other; /* When no echo-specific data is required */
      struct Pchoice_pet_r1{
         Pint      imp_dep;            /* impl. dependent*/
      } pet_r1;  /* For PET 1 */
      struct Pchoice_pet_r2 {
         Pint         num_prompts;     /* number of prompts          */
         Ppr_switch   *prompts;        /* array of prompts           */
         Pint         imp_dep;         /* impl. defined              */
      } pet_r2;       /* For PET 2 */
      struct Pchoice_pet_r3 {
         Pint      num_strings;        /* number of choice strings   */
         char      **strings;          /* array of choice strings    */
         Pint      imp_dep;            /* impl. defined              */
      } pet_r3;  /* For PET 3 */
      struct Pchoice_pet_r4 {
         Pint      num_strings;        /* number of alternatives     */
         char      **strings;          /* array of strings           */
         Pint      imp_dep;            /* impl. defined              */
      } pet_r4;  /* For PET 4 */
      struct Pchoice_pet_r5 {
         Pint      struct_id;          /* structure identifier       */
         Pint      num_pick_ids;       /* number of alternatives     */
         Pint      *pick_ids;          /* array of pick identifiers  */
         Pint      imp_dep;            /* impl. defined              */
      } pet_r5;  /* For PET 5 */
      Pint         imp_dep;            /* data for impl. defined pets */
   } pets;
   union Pchoice_measure_data {
      struct Pchoice_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    union Pchoice_trigger_data {
        struct Pchoice_trig_other {
            Pint        unused;
        } trig_other; /* When no trigger-specific data is required */
        Pint        imp_dep; /* data for impl. defined trigger processes */
    } trigger_data;
    union Pchoice_ack_data {
        struct Pchoice_ack_other {
            Pint        unused;
        } ack_other;  /* When no ack.-specific data is required */
        Pint        imp_dep; /* data for impl. defined ack. processes */
    } acknowledgement_data;
} Pchoice_data;

typedef struct {
    union Pchoice3_pets {
        struct Pchoice3_pet_other {
            Pint        unused;
        } pet_other; /* When no echo-specific data is required */
        struct Pchoice3_pet_r1{
            Pint        imp_dep;            /* impl. dependent              */
        } pet_r1;  /* For PET 1 */
        struct Pchoice3_pet_r2 {
            Pint        num_prompts;        /* number of prompts            */
            Ppr_switch*prompts;             /* array of prompts             */
            Pint        imp_dep;            /* impl. defined                */
        } pet_r2;       /* For PET 2 */
        struct Pchoice3_pet_r3 {
            Pint        num_strings;        /* number of choice strings     */
            char        **strings;          /* array of choice strings      */
            Pint        imp_dep;            /* impl. defined                */
        } pet_r3;  /* For PET 3 */
        struct Pchoice3_pet_r4 {
            Pint        num_strings;        /* number of alternatives       */
            char        **strings;          /* array of strings             */
            Pint        imp_dep;            /* impl. defined                */
        } pet_r4;  /* For PET 4 */
        struct Pchoice3_pet_r5 {
            Pint        struct_id;          /* structure identifier         */
            Pint        num_pick_ids;       /* number of alternatives       */
            Pint        *pick_ids;          /* array of pick identifiers    */
            Pint        imp_dep;            /* impl. defined                */
        } pet_r5;  /* For PET 5 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      Pint        imp_dep; /* data for impl. defined pets */
   } pets;
   union Pchoice3_measure_data {
      struct Pchoice_meas_other {
         Pint     unused;
      } meas_other; /* When no measure-specific data is required */
      Pint     imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Pchoice3_trigger_data {
      struct Pchoice3_trig_other {
         Pint     unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint     imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
   union Pchoice3_ack_data {
      struct Pchoice3_ack_other {
         Pint     unused;
      } ack_other;  /* When no ack.-specific data is required */
      Pint     imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pchoice_data3;

typedef struct {
   union Ploc_pets {
      struct Ploc_pet_other {
         Pint     unused;
      } pet_other;   /* When no echo-specific data is required */
      struct Ploc_pet_r1 {
         . . .                          /* impl. dependent          */
      } pet_r1;      /* Supports standard PET 1 */
      struct Ploc_pet_r2 {
         . . .                          /* impl. dependent          */
      } pet_r2;      /* Supports standard PET 2 */
      struct Ploc_pet_r3 {
         . . .                          /* impl. dependent          */
      } pet_r3;      /* Supports standard PET 3 */
      struct Ploc_pet_r4 {
         Pline_attrsline_attrs;         /* polyline attributes      */
         Pint     imp_dep;              /* impl. dependent          */
      } pet_r4;      /* Supports standard PET 4 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
      struct Ploc_pet_r5 {
         Pline_fill_ctrl_flagline_fill_ctrl_flag;/*ctrl. flag*/
         union Ploc_attrs {
            Pline_attrs   line_attrs; /* polyline attributes*/
            Pint_attrs    int_attrs;  /* interior attributes*/
            struct Ploc_fill_set {
               Pint_attrs  int_attrs;  /* interior attributes       */
               Pedge_attrs edge_attrs; /* edge attributes           */
            } fill_set;
         } attrs;
         Pint      imp_dep; /* data for impl. defined PET's */
      } pet_r5;       /* Supports standard PET 5 */
      struct Ploc_pet_r6 {
         . . .                          /* impl. dependent          */
      } pet_r6;       /* Supports standard PET 6 */
      Pint         imp_dep; /* data for impl. defined PET's */
   } pets;
   union Ploc_measure_data {
      struct Ploc_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Ploc_trigger_data{
      struct Ploc_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
   union Ploc_ack_data {
      struct Ploc_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Ploc_data;

typedef struct {
   union Ploc3_pets {
      struct Ploc3_pet_other {
         Pint      unused;
      } pet_other;   /* When no echo-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   struct Ploc3_pet_r1 {
      . . .                              /* impl. dependent              */
   } pet_r1;       /* Supports standard PET 1 */
   struct Ploc3_pet_r2 {
      . . .                              /* impl. dependent              */
   } pet_r2;       /* Supports standard PET 2 */
   struct Ploc3_pet_r3 {
      . . .                              /* impl. dependent              */
   } pet_r3;       /* Supports standard PET 3 */
   struct Ploc3_pet_r4 {
      Pline_attrs line_attrs;            /* polyline attributes          */
      Pint        imp_dep;               /* impl. dependent              */
   } pet_r4;       /* Supports standard PET 4 */
   struct Ploc3_pet_r5 {
      Pline_fill_ctrl_flag line_fill_ctrl_flag;  /* ctrl. flag        */
      union Ploc3_attrs {
         Pline_attrs   line_attrs; /* polyline attributes               */
         Pint_attrs    int_attrs;  /* interior attributes               */
         struct Ploc3_fill_set {
            Pint_attrs  int_attrs;   /* interior attributes             */
            Pedge_attrs edge_attrs;  /* edge attributes                 */
         } fill_set;
      } attrs;
      Pint      imp_dep; /* data for impl. defined PET's */
   } pet_r5;       /* Supports standard PET 5 */
   struct Ploc3_pet_r6 {
      . . .                              /* impl. dependent              */
   } pet_r6;       /* Supports standard PET 6 */
   Pint      imp_dep; /* data for impl. defined PET's */
} pets;
union Ploc3_measure_data {
   struct Ploc3_meas_other {
      Pint      unused;
   } meas_other;  /* When no measure-specific data is required */
   Pint      imp_dep; /* data for impl. defined measure processes */
} measure_data;
union Ploc3_trigger_data {
   struct Ploc3_trig_other {
      Pint      unused;
   } trig_other;  /* When no trigger-specific data is required */
   Pint      imp_dep; /* data for impl. defined trigger processes */
} trigger_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    union Ploc3_ack_data {
       struct Ploc3_ack_other {
          Pint      unused;
       } ack_other;    /* When no ack.-specific data is required */
       Pint      imp_dep; /* data for impl. defined ack. processes */
    } acknowledgement_data;

} Ploc_data3;

typedef struct {
    union Ppick_pets {
       struct Ppick_pet_other {
          Pint      unused;
       } pet_other;  /* When no echo-specific data is required */
       struct Ppick_pet_r1 {
          Pint      imp_dep;           /* impl. dependent            */
       } pet_r1;  /* For PET 1 */
       Pint      imp_dep; /* data for impl. defined pets */
       struct Ppick_pet_r2 {
          ...                          /* implementation dependent   */
       } pet_r2;  /* For PET 2 */
       struct Ppick_pet_r3 {
          ...                          /* implementation dependent   */
       } pet_r3;  /* For PET 3 */
       struct Ppick_pet_r4 {
          Pint      highl_ind;         /* highlighting index         */
          Pint      imp_dep;           /* impl. dependent            */
       } pet_r4;  /* For PET 4 */
       struct Ppick_pet_r5 {
          Pint      highl_ind;         /* highlighting index         */
          Pint      imp_dep;           /* impl. dependent            */
       } pet_r5;  /* For PET 5 */
       struct Ppick_pet_r6 {
          Pint      highl_ind;         /* highlighting index         */
          Pint      imp_dep;           /* impl. dependent            */
       } pet_r6;  /* For PET 6 */
       Pint         imp_dep;           /* implementation defined PET's*/
    } pets;
    union Ppick_measure_data {
       struct Ppick_meas_other {
          Pint      unused;
       } meas_other; /* When no measure-specific data is required */
       Pint      imp_dep; /* data for impl. defined measure processes */
    } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Ppick_trigger_data {
      struct Ppick_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined pick processes */
   } trigger_data;
   union Ppick_ack_data {
      struct Ppick_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Ppick_data;

typedef struct {
   union Ppick3_pets {
      struct Ppick3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Ppick3_pet_r1 {
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r1;  /* For PET 1 */
      struct Ppick3_pet_r2 {
         ...                           /* implementation dependent     */
      } pet_r2;  /* For PET 2 */
      struct Ppick3_pet_r3 {
         ...                           /* implementation dependent     */
      } pet_r3;  /* For PET 3 */
      struct Ppick3_pet_r4 {
         Pint      highl_ind;          /* highlighting index           */
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r4;  /* For PET 4 */
      struct Ppick3_pet_r5 {
         Pint      highl_ind;          /* highlighting index           */
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r5;  /* For PET 5 */
      struct Ppick3_pet_r6 {
         Pint      highl_ind;          /* highlighting index           */
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r6;
      Pint      imp_dep; /* data for impl. defined pets */
   } pets;  /* For PET 6 */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    union Ppick3_measure_data {
       struct Ppick3_meas_other {
          Pint      unused;
       } meas_other; /* When no measure-specific data is required */
       Pint      imp_dep; /* data for impl. defined measure processes */
    } measure_data;
    union Ppick3_trigger_data {
       struct Ppick3_trig_other {
          Pint      unused;
       } trig_other; /* When no trigger-specific data is required */
       Pint      imp_dep; /* data for impl. defined pick processes */
    } trigger_data;
    union Ppick3_ack_data {
       struct Ppick3_ack_other {
          Pint      unused;
       } ack_other; /* When no ack.-specific data is required */
       Pint      imp_dep; /* data for impl. defined ack. processes */
    } acknowledgement_data;
} Ppick_data3;

 typedef struct {
    Pint              in_buf_size;    /* input buffer size          */
    Pint              init_pos;       /* initial editing position   */
    union Pstring_pets {
       struct Pstring_pet_other {
          Pint      unused;
       } pet_other;  /* When no echo-specific data is required */
       struct Pstring_pet_r1{
          Pint      imp_dep;          /* impl. dependent            */
       } pet_r1;  /* For PET 1 */
       Pint      imp_dep; /* data for impl. defined pets */
    } pets;
    union Pstring_measure_data {
       struct Pstring_meas_other {
          Pint      unused;
       } meas_other; /* When no measure-specific data is required */
       Pint      imp_dep; /* data for impl. defined measure processes */
    } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pstring_trigger_data {
      struct Pstring_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
   union Pstring_ack_data {
      struct Pstring_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pstring_data;

typedef struct {
   Pint              in_buf_size;    /* input buffer size          */
   Pint              init_pos;       /* initial editing position   */
   union Pstring3_pets {
      struct Pstring_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstring3_pet_r1{
         Pint      imp_dep;          /* impl. dependent            */
      } pet_r1;  /* For PET 1 */
      Pint      imp_dep; /* data for impl. defined pets */
   } pets;
   union Pstring3_measure_data {
      struct Pstring3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Pstring3_trigger_data {
      struct Pstring3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pstring3_ack_data {
      struct Pstring3_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pstring_data3;

typedef struct {
   Pint              buffer_size;    /* input buffer size */
   Pint              init_pos;       /* initial editing position */
   Pfloat            x_interval;     /* x trigger interval */
   Pfloat            y_interval;     /* y trigger interval */
   Pfloat            time_interval; /* time trigger interval */
   union {
      struct Pstroke_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstroke_pet_r1 {
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r1;  /* For std PET 1 */
      struct Pstroke_pet_r2 {
         Pint      imp_dep;            /* impl. dependent              */
      } pet_r2;  /* For std PET 2 */
      struct Pstroke_pet_r3 {
         Pmarker_attrs  marker_attrs; /* marker attributes            */
         Pint           imp_dep;      /* impl. dependent              */
      } pet_r3;  /* For std PET 3 */
      struct Pstroke_pet_r4 {
         Pline_attrs line_attrs;      /* line attributes              */
         Pint        imp_dep;         /* impl. dependent              */
      } pet_r4;  /* For std PET 4 */
      Pint      imp_dep; /* data for impl. defined PET's */
         } pets;
   union Pstroke_measure_data {
      struct Pstroke_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pstroke_trigger_data {
      struct Pstroke_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
   union Pstroke_ack_data {
      struct Pstroke_ack_other {
         Pint      unused;
      } ack_other;  /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;
} Pstroke_data;

typedef struct {
   Pint              in_buf_size;    /* input buffer size          */
   Pint              init_pos;       /* initial editing position   */
   Pfloat            x_interval;     /* x trigger interval         */
   Pfloat            y_interval;     /* y trigger interval         */
   Pfloat            z_interval;     /* z trigger interval         */
   Pfloat            time_interval; /* time trigger interval       */
   union Pstroke3_pets {
      struct Pstroke3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pstroke3_pet_r1 {
         Pint      imp_dep;           /* impl. dependent           */
      } pet_r1;  /* For std PET 1 */
      struct Pstroke3_pet_r2 {
         Pint      imp_dep;           /* impl. dependent           */
      } pet_r2;  /* For std PET 2 */
      struct Pstroke3_pet_r3 {
         Pmarker_attrs  marker_attrs; /* marker attributes         */
         Pint           imp_dep;      /* impl. dependent           */
      } pet_r3;  /* For std PET 3 */
      struct Pstroke3_pet_r4 {
         Pline_attrs   line_attrs;    /* line attributes           */
         Pint          imp_dep;       /* impl. dependent           */
      } pet_r4;  /* For std PET 4 */
      Pint      imp_dep; /* data for impl. defined PET's */
   } pets;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    union Pstroke3_measure_data {
       struct Pstroke3_meas_other {
          Pint      unused;
       } meas_other; /* When no measure-specific data is required */
       Pint      imp_dep; /* data for impl. defined measure processes */
    } measure_data;
    union Pstroke3_trigger_data {
       struct Pstroke3_trig_other {
          Pint      unused;
       } trig_other; /* When no trigger-specific data is required */
       Pint      imp_dep; /* data for impl. defined trigger processes */
    } trigger_data;
    union Pstroke3_ack_data {
       struct Pstroke3_ack_other {
          Pint      unused;
       } ack_other;  /* When no ack.-specific data is required */
       Pint      imp_dep; /* data for impl. defined ack. processes */
    } acknowledgement_data;
} Pstroke_data3;

typedef struct {
    Pfloat              low;          /* Low value of valuator range */
       Pfloat      high;              /* High value of valuator range*/
    union Pval_pets {
       struct Pval_pet_other {
          Pint      unused;
       } pet_other;  /* When no echo-specific data is required */
       Pint      imp_dep; /* data for impl. defined pets */
    } pets;
    union Pval_measure_data {
       struct Pval_measure_other {
          Pint      unused;
       } meas_other; /* When no measure-specific data is required */
       Pint      imp_dep; /* data for impl. defined measure processes */
    } measure_data;
    union Pval_trigger_data {
       struct Pval_trig_other {
          Pint      unused;
       } trig_other; /* When no trigger-specific data is required */
       Pint      imp_dep; /* data for impl. defined trigger processes */
    } trigger_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   union Pval_ack_data {
      struct Pval_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pval_data;

typedef struct {
   Pfloat             low;          /* Low value of valuator range */
      Pfloat       high;            /* High value of valuator range*/
   union Pval3_pets {
      struct Pval3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      Pint      imp_dep; /* data for impl. defined pets */
   } pets;
   union Pval3_measure_data {
      struct Pval3_measure_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Pval3_trigger_data {
      struct Pval3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined trigger processes */
   } trigger_data;
   union Pval3_ack_data {
      struct Pval3_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pval_data3;"
```

Page 194 (Page 105 Amendment 1)

The following text replaces the definition of Pcolr_rep_ptr*:*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

"

```
typedef union {

   /* start of PHIGS PLUS union members */
   Prgb       *rgb;          /* pointer to RGB colour values */
   Pcieluv    *cieluv;       /* pointer to CIELUV pointer values */
   Phls       *hls;          /* pointer to HLS pointer values */
   Phsv       *hsv;          /* pointer to HSV pointer values */
   /* end of PHIGS PLUS union members */
   /* start of Full PHIGS union members */
   Prgba      *rgba;         /* pointer to RGBA colour values */
   Pcieluva   *cieluva;      /* pointer to CIELUVA pointer values */
   Phlsa      *hlsa;         /* pointer to HLSA pointer values */
   Phsva      *hsva;         /* pointer to HSVA pointer values */
   /* end of Full PHIGS union members */
   ...                       /* implementation defined */

} Pcolr_rep_ptr;"
```

Page 194 (Page 118 Amendment 1)

The following text replaces the definition of Ppat_rep_plus*:*

"

```
typedef struct {

   Pint           colr_type;          /* colour type                */
   Pcolrv_array   colrs;              /* colour array               */

} Pgcolr_array;

typedef Pgcolr_array Ppat_rep_plus;

typedef struct {

   Pint           colr_type;          /* colour type                */
   Pint           num_arrays;         /* number of arrays in set    */
   Pcolrv_array   *colrs;             /* set of colour arrays       */

} Pgcolr_array_set;
```

Page 194 (Page 122 Amendment 1)

The following text is inserted after the definition of Pws_st_tables_plus:

"

```
typedef struct {

   Pdyn_mod          posting_status;/* posting status modification */

} Pdyns_post_grps;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {

 PACCESS_REFERENCED = 0,
 PACCESS_COPIED = 1

} Paccess_flag;

typedef enum {

   P_ACK_ACCEPTANCE = 0,
   P_ACK_NONACCEPTANCE = 1

} Packnowledgement_type;

typedef struct {

 Pack_type        type;
 Pint             process_id;

} Pack_process;

typedef struct {

 Pint             count;
 Pack_process     *procs;

} Pack_process_list;

typedef enum {

 PNOTASSOCIATED = 0,
 PASSOCIATED = 1

} Passoc_flag;

typedef struct {

   Pint      max_simul_trig,   /* max simul. trigger procs           */
   Pint      max_simul_echo,   /* max simul. echo procs              */
   Pint      max_simul_ack,    /* max simul. acknowledgement procs   */
   Pint_list meas_proc_ids,    /* available measure process ids      */
   Pint_list trig_proc_ids,    /* available trigger process ids      */
   Pint_list echo_proc_ids,    /* available echo process ids         */
   Pint_list ack_proc_ids      /* available acknowledgement          */
                               /* process ids                        */

} Patomic_lid_facs;

typedef enum {

   PBACKG_REDISPLAY_OFF = 0,
   PBACKG_REDISPLAY_ON = 1

} Pbackg_redisplay;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
  PBACKG_STYLE_TRANSPARENT = 0,
  PBACKG_STYLE_OPAQUE = 1
} Pbackg_style;

typedef enum {
  PFALSE = 0,
  PTRUE = 1
} Pboolean;

typedef enum {
  PBORDER_OFF = 0,
  PBORDER_ON = 1
} Pborder_indic;

typedef enum {
  PCLIPCOND_IS_NOT_CLIPPED = 0,
  PCLIPCOND_IS_PARTIALLY_CLIPPED = 1,
  PCLIPCOND_IS_FULLY_CLIPPED = 2
} Pclip_cond;

typedef enum {
   PCLOSURE_PIE = 0,
   PCLOSURE_SEGMENT = 1
} Pclosure;

union _Pin_class_data;

typedef struct _Pset_data {
   Pin_class        dev_class;
   Pint             num_in_set;
   _Pin_class_data  *data;
} Pset_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

   Pint      max_measures,              /* max instances                */
   Pint      max_simul_trig,            /* max simul. trigger procs     */
   Pint      max_simul_echo,            /* max simul. echo procs        */
   Pint      max_simul_ack,             /* max simul. ack. procs        */
   Pint_list loc_meas_proc_ids,         /* avail locator meas proc ids */
   Pint_list stroke_meas_proc_ids,      /* avail stroke meas proc ids  */
   Pint_list val_meas_proc_ids,         /* avail valuator meas proc ids*/
   Pint_list choice_meas_proc_ids,      /* avail choice meas proc ids  */
   Pint_list pick_meas_proc_ids,        /* avail pick meas proc ids    */
   Pint_list string_meas_proc_ids,      /* avail string meas proc ids  */
   Pint_list set_meas_proc_ids,         /* avail set meas proc ids     */
   Pint_list composite_meas_proc_ids, /* avail compos meas proc ids  */
   Pint_list trig_proc_ids,             /* avail trigger proc ids      */
   Pint_list loc_echo_proc_ids,         /* avail locator echo proc ids */
   Pint_list stroke_echo_proc_ids,      /* avail stroke echo proc ids  */
   Pint_list val_echo_proc_ids,         /* avail valuator echo proc ids*/
   Pint_list choice_echo_proc_ids,      /* avail choice echo proc ids  */
   Pint_list pick_echo_proc_ids,        /* avail pick echo proc ids    */
   Pint_list string_echo_proc_ids,      /* avail string echo proc ids  */
   Pint_list set_echo_proc_ids,         /* avail set echo proc ids     */
   Pint_list comp_echo_proc_ids,        /* avail compos echo proc ids  */
   Pint_list ack_proc_ids               /* avail ack. proc ids         */

} Pset_facs;

typedef struct _Pcomposite_individual {

   Pin_class        class;
   _Pin_class_data  *data;

} Pcomposite_individual;

typedef struct {

   Pint                    count;
   Pcomposite_individual   *individual;

} Pcomposite_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
    Pint      max_simul_meas,        /* max simul. measure procs       */
    Pint      max_simul_trig,        /* max simul. trigger procs       */
    Pint      max_simul_echo,        /* max simul. echo procs          */
    Pint      max_simul_ack,         /* max simul. acknowledgement procs
                                        */
    Pint_list loc_meas_proc_ids,     /* avail. locator measure proc. ids
                                        */
    Pint_list stroke_meas_proc_ids, /* avail. stroke measure proc. ids
                                        */
    Pint_list val_meas_proc_ids,     /* avail. valuator measure proc. ids
                                        */
    Pint_list choice_meas_proc_ids, /* avail. choice measure proc. ids
                                        */
    Pint_list pick_meas_proc_ids,    /* avail. pick measure proc. ids  */
    Pint_list string_meas_proc_ids, /* avail. string measure proc. ids
                                        */
    Pint_list set_meas_proc_ids,     /* avail. set measure proc. ids   */
    Pint_list composite_meas_proc_ids,/*avail. composite meas. proc. ids
                                        */
    Pint_list trig_proc_ids,         /* available trigger process ids  */
    Pint_list loc_echo_proc_ids,     /* available locator echo proc. ids
                                        */
    Pint_list stroke_echo_proc_ids, /* available stroke echo proc. ids
                                        */
    Pint_list val_echo_proc_ids,     /* available valuator echo proc. ids
                                        */
    Pint_list choice_echo_proc_ids, /* available choice echo proc. ids
                                        */
    Pint_list pick_echo_proc_ids,    /* available pick echo proc. ids  */
    Pint_list string_echo_proc_ids, /* available string echo proc. ids
                                        */
    Pint_list set_echo_proc_ids,     /* available set echo proc. ids   */
    Pint_list comp_echo_proc_ids,    /* available composite echo proc. ids
                                        */
    Pint_list ack_proc_ids           /* available ack. proc. ids       */
} Pcomposite_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef union _Pin_class_data {

   Pint                unused;
   Ploc_data           loc;
   Ploc_data3          loc3;
   Pstroke_data        stroke;
   Pstroke_data3       stroke3;
   Pval_data           val;
   Pval_data3          val3;
   Pchoice_data        choice;
   Pchoice_data3       choice3;
   Ppick_data          pick;
   Ppick_data3         pick3;
   Pstring_data        string;
   Pstring_data3       string3;
   Pset_data           set;
   Pcomposite_data     composite;
   ...                                  /* other classes of input data */

} Pin_class_data;

typedef enum {

  PCOMP_IN_STATUS_NONE = 0,
  PCOMP_IN_STATUS_OK = 1,
  PCOMP_IN_STATUS_INCOMPLETE = 2

} Pcomp_in_style;

typedef struct {

   Pcond_flag_mask     enable_mask;    /* condition flag enable mask  */
   Pcond_flag_mask     disable_mask;   /* condition flag disable mask */

} Pcond_flags;

typedef struct {

   Ptest               test;           /* test operation              */
   Pcond_flags         mask;           /* condition test mask         */

} Pcond_flag_test;

typedef struct {

   Pint                num_tests;      /* number of condition test    */
   Pcond_flag_test     *tests;         /* list of condition flag tests*/

} Pcond_test_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   Pint       num_invis_filters;/* number of avail. invisibility filters
                                */
   Pint       num_highl_filters;/* number of avail. highlighting filters
                                */
   Pint       num_pick_filters; /* number of avail. pick filters     */
   Pint       num_appl_filters; /* number of avail. application filters
                                */
   Pint       num_appl_integers;/* number of application integer     */
                               /* values supported                  */
   Pint       num_appl_reals;   /* number of application real values*/
                               /* supported                         */
   Pint_list  test_methods;     /* list of test methods available    */
} Pcond_trav_facs;

typedef struct {
   Pint            name_size;       /* size of name                 */
   char            *name;           /* configuration setting name   */
} Pconfig_name;

typedef struct {
   Pint            list_size;       /* number of names              */
   Pconfig_name    *name;           /* configuration name list      */
} Pconfig_name_list;

typedef struct {
   Pint          num_supported;   /* number of simultaneously supported
                                     */
                                  /* device coordinate clip regions */
} Pconfig_setting_facs;

typedef union {
   Pint        int_value;          /* integer configuration value  */
   Pfloat      real_value;         /* real configuration value     */
   char        *string_value;      /* string configuration value   */
} Pconfig_value;

typedef struct {
   char            *name;           /* configuration setting name   */
   Pconfig_value   *value;          /* configuration setting value  */
} Pconfig_settings;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

   Pint      num_supported;    /* number of simultaneously supported */
                               /* device coordinate clip regions     */

} Pdc_clip_facs;

typedef struct {

   unsigned int mode:1;
   unsigned int ref_planes:1;
   unsigned int scale:1;
   unsigned int gcolr:1;
   unsigned int :28;

} Pdepth_cue_mask;

typedef enum {

   PDIM_INTERPRET_NONE = 0,
   PDIM_INTERPRET_STATE = 1,
   PDIM_INTERPRET_ALL = 2

} Pdi_mode;

typedef struct {

   Pint   num_trav_procs;         /* number of simultaneously posted */
                                  /* traversal processes             */
   Pint   num_rend_algorithms;    /* number of multi-pass algorithms */
                                  /* supported for rendering         */
   Pint   num_pick_algorithms;    /* number of multi-pass algortihms */
                                  /* supported for picking           */

} Pdi_trav_facs;

typedef struct {

  Pin_class       dev_class;       /* input class*/
  Pint            instance;        /* i = ith of class in LID def.*/

} Pecho_measure_proc;

typedef struct {

  Pint                pet;
  Pecho_measure_proc *measures_echoed;
  Pint                process_id;

} Pecho_process;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {

  Pint              count;
  Pecho_process     *procs;

} Pecho_process_list;

typedef struct {

    Pedge_flag          flag;          /* edge flag                    */
    Pint                type;          /* edgetype                     */
    Pfloat              width;         /* edgewidth scale factor       */
    Pgolr               colr;          /* edge colour                  */
    Plinetype_adapt     adapt;         /* edgetype adaptability        */
    Plinetype_cont      cont;          /* edgetype continuity          */
    Pfloat              offset;        /* edgetype offset              */
    Plinecap            cap;           /* edgecap                      */
    Plinejoin           join;          /* edgejoin                     */
    Pfloat              limit;         /* edgemitre limit              */

} Pedge_bundle_full;

typedef struct {

    unsigned int flag:1;
    unsigned int type:1;
    unsigned int width:1;
    unsigned int colr:1;
    unsigned int adapt:1;
    unsigned int cont:1;
    unsigned int offset:1;
    unsigned int cap:1;
    unsigned int join:1;
    unsigned int limit:1;
    unsigned int :22;

} Pedge_mask;

typedef enum {

    PEDTA_EXACT = 0,
    PEDTA_ADAPT = 1

} Pedgetype_adapt;

typedef enum {

    PEDC_BUTT = 0,
    PEDC_ROUND = 1,
    PEDC_SQUARE = 2

} Pedgecap;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef enum {

   PEDTC_CONTINUOUS = 0,
   PEDTC_RESTART = 1

} Pedgetype_cont;

typedef enum {

   PEDJ_FLAT = 0,
   PEDJ_MITRE = 1,
   PEDJ_ROUND = 2,
   PEDJ_BEVEL = 3

} Pedgejoin;

typedef struct {

   Pint_list pattern_types;      /* list of supported pattern types   */
   Pint_list image_spec_methods; /* list of supported image           */
                                 /* specifcation methods              */

} Pext_pat_facs;

typedef enum {

   PERASE_OFF = 0,
   PERASE_ON = 1

} Perase;

typedef struct {

   Pint_list image_spec_methods; /* list of supported image           */
                                 /* specification methods             */
   Pint      num_def_ids;        /* number of definable image         */
                                 /* resource identifiers              */
   Pint      num_predef_ids;     /* number of predefined image        */
                                 /* resources                         */

} Pimage_res_facs;

typedef struct {

   unsigned int style:1;
   unsigned int style_ind:1;
   unsigned int colr:1;
   unsigned int shad_method:1;
   unsigned int :28;

} Pinterior_mask;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint              num_limits;         /* number of limits          */
  Plimit            *limits;            /* list of limits            */
} Plimit_list;

typedef struct {
  Pint              num_limits;         /* number of limits          */
  Plimit3           *limits;            /* list of limits            */
} Plimit3_list;

typedef struct {
   Pint     light_ws_id;                /* workstation id for view   */
   Pint     light_src_ind;              /* view index                */
   Pint     light_action_type;          /* view action type          */
   Pint     transfer_type;              /* measure transfer type     */
} Plight_attach;

typedef enum {
   PLNC_BUTT = 0,
   PLNC_ROUND = 1,
   PLNC_SQUARE = 2
} Plinecap;

typedef enum {
   PLNJ_FLAT = 0,
   PLNJ_MITRE = 1,
   PLNJ_ROUND = 2,
   PLNJ_BEVEL = 3
} Plinejoin;

typedef enum {
   PLNTA_EXACT = 0,
   PLNTA_ADAPT = 1
} Plinetype_adapt;

typedef enum {
   PLNTC_CONTINUOUS = 0,
   PLNTC_RESTART = 1
} Plinetype_cont;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {

   Pint                type;           /* linetype                      */
   Pfloat              width;          /* linewidth scale factor        */
   Pgolr               colr;           /* polyline colour               */
   Pint                shad_method;    /* polyline shading method       */
   Pcurve_approx_crit_data
                       curve_approx_crit_data;
                                       /* curve approx. criteria        */
   Plinetype_adapt     adapt;          /* linetype adaptability         */
   Plinetype_cont      cont;           /* linetype continuity           */
   Pfloat              offset;         /* linetype offset               */
   Plinecap            cap;            /* linecap                       */
   Plinejoin           join;           /* linejoin                      */
   Pfloat              limit;          /* linemitre limit               */

} Pline_bundle_full;

typedef struct {

   Pint      num_definable;       /* number definable linetypes       */
   Pint      num_predefined;      /* number predefined linetypes      */
   Pint      max_dash_segments;   /* maximum dash segments supported */
   Pfloat    offset;              /* offset to definable linetypes    */
   Pint_list predef_def_linetypes /* list of predef. definable linetypes
                                    */

} Plinetype_def_facs;

typedef struct {

   Pfloat           lum_value;        /* luminance_value              */
   Pfloat           alpha_value;      /* alpha value                  */

} Plum_alpha_pair;

typedef struct {

   Pint_size        dims;         /* array dimensions                 */
   Plum_alpha_pair  *values;      /* 2D array of luminance/alpha pairs */

} Plum_alpha_array;

typedef struct {

   Pint             num_arrays;  /* number of arrays in set          */
   Plum_alpha_array *sets;       /* set of 2D luminance/alpha arrays */

} Plum_alpha_array_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   Pint                shape_type;     /* shape type                    */
   union _Pshape_data {
      Ppoint_list polyline;           /* for PSHAPE_POLYLINE           */
      Ppoint_list fill_area;          /* for PSHAPE_FILL_AREA          */
      Ppoint_list convex_fill_area;   /* for PSHAPE_CONVEX_FILL_AREA */
      ...                             /* implementation defined        */
   } shape_data;                      /* marker shape data             */
} Pmarker_data;

typedef struct {
   Pint            number_shapes;     /* number of shapes              */
   Pmarker_data    *marker;           /* array of shape descriptors    */
} Pmarker_desc;

typedef struct {
   Pint       num_definable;       /* number definable marker types  */
   Pint       num_predefined;      /* number predefined marker types */
   Pfloat     offset;              /* offset to definable linetypes  */
   Pint       max_allowable_pts;   /* maximum allowable points       */
   Pint_list  shape_types;         /* list of available shape types  */
   Pint_list  predef_marker_types  /* list of predefined marker types
                                      */
} Pmarker_type_def_facs;

typedef union _Pin_class_measure;/*  used for defining _Pmeasure    */
                                 /*  and _Pset_measure              */
typedef struct _Pmeasure {
   Pin_class          class;
   _Pin_class_measure *data;
} Pmeasure;

typedef struct _Pset_measure {
   Pin_class          dev_class;      /* input class of set          */
   Pint               set_size;       /* size of measure set         */
   _Pin_class_measure *measure_set;   /* array of measures of same   */
                                      /* class                       */
} Pset_measure;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct _Pcomposite_measure {

   Pint         count;                /* number of measure components*/
   Pmeasure     *measures;            /* list of measure components  */

} Pcomposite_measure;


typedef union _Pin_class_measure {

   struct {
      Pint         view_ind;
      Ppoint       loc_pos;
   } loc;

   struct {
      Pint         view_ind;
      Ppoint3      loc_pos;
   } loc3;

   struct {
      Pint         view_ind;
      Ppoint_list  points;
   } stroke;

   struct {
      Pint         view_ind;
      Ppoint_list3 points;
   } stroke3;

   Pfloat              val;

   struct {
      Pin_status   istat;
      Pint         selection;
   } choice;

   struct {
      Pin_status   istat;
      Ppick_path   *path;
   } pick;

   char                 *string;

   Pset_measure         *set;

   Pcomposite_measure   *composite;

   ...                 /* other classes which may be added */

} Pin_class_measure;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

  Pin_class        dev_class;
  Pint             process_id;

} Pmeas_process;

typedef struct _Pmeas_process_list {

   Pint              count;          /* number of process in list   */
   Pmeas_process     *measure_procs; /* list of measure processes   */

} Pmeas_process_list;

typedef struct {

  Pint             set_maxsize;       /* maximum set size            */
  Pmeas_process    process_id;        /* measure process identifier  */

} Pmeas_process_set;

typedef enum {

   PNA_IN_STATUS_NONE = 0,
   PNA_IN_STATUS_INCOMPLETE = 1,
   PNA_IN_STATUS_OK = 2

} Pna_in_status;

typedef struct {

   Pint         set;             /* number available set LIDs        */
   Pint         composite;       /* number available composite LIDs  */

} Pnum_na_in;

typedef struct {

   unsigned int pat_bundle:1;
   unsigned int pat_rep_plus:1;
   unsigned int ext_pat_rep:1;
   unsigned int :29

} Ppattern_mask;

typedef struct _Ppick_path_list {

   Pint         num_paths;       /* number paths in list             */
   Ppick_path   *paths;          /* array of pick paths              */

} Ppick_path_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   Pint                 type;          /* pattern type */
   union _Ppattern_data {
      struct Ppat_colr_ind_array {
         Ppat_rep  pat_rep;            /* pattern representation      */
      } pat_colr_ind_array;
      struct Ppat_colr_array {
         Ppat_rep_pluspat_rep_plus;    /* pattern representation plus */
      } pat_colr_array;
      struct Ppat_image_res {
         Pint      image_res_id;       /* pattern image resource      */
      } pat_image_res;
      ...                              /* implementation defined      */
   } data;                             /* pattern data */
} Ppattern;

typedef enum {
   PPICK_STAT_FALSE = 0,
   PPICK_STAT_TRUE = 1
} Ppick_stat;

typedef struct {
   Pint                 pick_type;
   union _Ppick_type_data {
      Pint         unused;             /* workstation-dependent       */
      Pfloat_size3 rect_aperture;      /* 3D rectangular aperture     */
      Pfloat_size3 vis_rect_aperture;  /* 3D rectangular aperture     */
                                       /* visible primitives only     */
      Ppoint_list  user_aperture;      /* user-specified aperture     */
      . . .        /* data for impl. defined pick types */
   } data;
} Ppick_type_data;

typedef enum {
   PPICT_STAT_COMP = 0,
   PPICT_STAT_INCOMP = 1
} Ppict_stat;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

   unsigned int type:1;
   unsigned int width:1;
   unsigned int colr:1;
   unsigned int shad_method:1;
   unsigned int curve_approx_crit_data:1;
   unsigned int adapt:1;
   unsigned int cont:1;
   unsigned int offset:1;
   unsigned int cap:1;
   unsigned int join:1;
   unsigned int limit:1;
   unsigned int :21;

} Ppolyline_mask;

typedef struct {

   unsigned int type:1;
   unsigned int size:1;
   unsigned int colr:1;
   unsigned int :29;

} Ppolymarker_mask;

typedef enum {

   PPOSTED_FALSE = 0,
   PPOSTED_TRUE = 1

} Pposted_ind;

typedef struct {

   Pint      num_def_grps;    /* number of definable posting groups  */
   Pint      num_predef_grps; /* number of predefined posting groups */
   Pint_list predef_grps;     /* list of predefined posting groups   */
   Pint_list backg_methods;   /* list of supported posting group     */
                              /* background methods                  */

} Ppost_grp_facs;

typedef enum {

   PPGSTAT_INACTIVE = 0,
   PPGSTAT_ACTIVE = 1

} Ppost_grp_status;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

   unsigned int refl_model:1;
   unsigned int data:1;
   unsigned int :30;

} Prefl_mask;

typedef enum {

   PREF_EXECUTE = 0,
   PREF_INSTANCE = 1

} Pref_type;

typedef enum {

  PSCALE_NOT_SCALABLE,
  PSCALE_SCALABLE

} Pscale_flag;

typedef enum {

   PSI_NOT_SUPPORTED = 0,
   PSI_SUPPORTED = 1

} Psupport_indication;

typedef enum {

   PTARG_DATA_ST_ABSENT = 0,
   PTARG_DATA_ST_PRESENT = 1

} Ptarg_data_st;

typedef enum {

   PTARG_EMPTY_ST_EMPTY = 0,
   PTARG_EMPTY_ST_NOT_EMPTY = 1

} Ptarg_empty_st;

typedef struct {

   Pint           num_avail_targets; /* number of available targets */

} Ptarget_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
   PBASE_TARG = 0,
   PREND_TARG = 1,
   PDISP_TARG = 2
} Ptarg_type;

typedef struct {
   Pint              num_targ_types;/* number of target types      */
   Ptarg_type        *targ_types;   /* list of target types        */
} Ptarg_type_list;

typedef struct {
 Ptarg_type      ref_targ;          /* reference target            */
 Pint            offset;            /* offset from reference target*/
} Ptarg_addr;

typedef struct {
     Pint        type;              /* target operation type       */
   union _Ptarget_op_data {
      struct Prend_targ {
         Ptarg_addrtarg_addr;       /* target address              */
      } rend_targ;
      struct Pdisp_targ {
         Ptarg_addrtarg_addr;       /* target address              */
      } disp_targ;
      struct Pclear_targ {
         Ptarg_addrtarg_addr;       /* target address              */
      } clear_targ;
      struct Pcopy_targ {
         Ptarg_addrsrc_targ_addr;   /* source target address       */
         Ptarg_addrdest_targ_addr;  /* destination target address  */
      } copy_targ;
      ...                           /* implementation defined      */
   } data;                          /* target operation data       */
} Ptarg_op;

typedef struct {
 Pint            num_ops;           /* number of target operations */
 Ptarg_op        *targ_ops;         /* list of target operations   */
} Ptarg_op_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {

   PTRL_EQUAL = 0,
   PTRL_NOT_EQUAL = 1,
   PTRL_GREATER = 2,
   PTRL_LESS = 3,
   PTRL_GREATER_OR_EQUAL = 4,
   PTRL_LESS_OR_EQUAL = 5

} Ptest_comp_rel;

typedef enum {

   PTLG_BITWISE_AND = 1,
   PTLG_BITWISE_OR = 2,
   PTLG_BITWISE_XOR = 3

} Ptest_comp_logical;

typedef struct {

   unsigned int font:1;
   unsigned int precision:1;
   unsigned int char_expan:1;
   unsigned int char_space:1;
   unsigned int colr:1;
   unsigned int :27;

} Ptext_mask;

typedef struct {

  Pint            id;             /* traversal resource identifier*/
  Pint            type;           /* traversal resource type*/

} Ptrav_res;

typedef struct {

  Pint            num_trav_res;   /* number of traversal resources*/
  Ptrav_res       *trav_res;      /* list of traversal resources*/

} Ptrav_res_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   Ptrav_res_list avail_trav_res;/* list of avail trav. resources   */
} Ptrav_res_facs;

typedef enum {
   PTRAV_RENDER = 0,
   PTRAV_PICK = 1
} Ptrav_type;

typedef enum {
   PTRIGGER_BREAK = 0,
   PTRIGGER_SELECT = 1,
   PTRIGGER_EVENT = 2,
   PTRIGGER_DELETE = 3
} Ptrig_type;

typedef struct {
  Ptrig_type       type;
  Pint             process_id;
} Ptrig_process;

typedef struct {
  Pint             count;
  Ptrig_process    *procs;
} Ptrig_process_list;

typedef struct {
   Pint     view_ws_id;              /* workstation id for view     */
   Pint     view_ind;                /* view index                  */
   Pint     view_action_type;        /* view action type            */
   Pint     transfer_type;           /* measure transfer type       */
   Pint     view_upd_method;         /* view update method          */
} Pview_attach;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
   PVIEW_ST_CORRESPOND = 0,
   PVIEW_ST_DIFFERENT = 1
} Pview_status;

typedef enum {
   PWATCH_OFF = 0,
   PWATCH_ON = 1
} Pwatch_enable;"

typedef struct {
 Pint_size        dims;              /* float array dimensions      */
 Pfloat           *floats;           /* float array                 */
} Pfloat_array;

typedef struct {
 Pint             num_arrays;      /* number of float arrays in set*/
 Pfloat_array     *floats;         /* float arrays*/
} Pfloat_array_set;

typedef struct {
   Pint image_spec_method;
   union _Pimage_data {
      struct Puncomp_colr_ind_array {
         Pint      *colr_ind_array;   /* uncompressed array of       */
                                      /* colour indices              */
      } uncomp_colr_ind_array;
      struct Puncomp_colr_array {
         Pgcolr_array  *colr_array;   /* uncompressed array of       */
                                      /*  colours                    */
      } uncomp_colr_array;
      struct Pwindow_sys_bitmap {
         Pint          handle;        /* bitmap handle               */
      } window_sys_bitmap;
      struct Pvideo_sig_chan_id {
         Pint          chan_id;       /* video signal channel id     */
      } video_sig_chan_id;
      struct Pluminance {
         Pfloat_array  *lum_values;   /* uncompressed array of*/
                                      /* luminance values*/
      } luminance;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Pluminance_alpha {
         Plum_alpha_array  *lum_values;  /* uncompressed array of     */
                                         /* luminance values          */
      } luminance_alpha;
      struct Pmipmap_colrs {
         Pgcolr_array_set  *colr_arrays; /* set of uncompressed       */
                                         /* colour arrays             */
      } mipmap_colrs;
      struct Pmipmap_luminance {
         Pfloat_array_set  *lum_values;  /* set of uncompressed       */
                                         /* luminance arrays          */
      } mipmap_luminance;
      struct Pmipmap_luminance_alpha {
         Plum_alpha_array_set *lum_values;  /* set of uncompressed    */
                                            /* luminance/alpha arrays */
      } mipmap_luminance_alpha;
      int       impl_def            /* implementation defined methods */
   } image_data;

} Pimage_res;

typedef struct {
   Pint                 method;         /* background method          */
   union Pbackg_method_data {
      struct Pcolr_table_0 {
         Pint      unused;              /* unused                     */
      } colr_table_0;
      struct Pcolr_ind {
         Pint      colr_ind;            /* colour index               */
      } colr_ind;
      struct Pcolr {
         Pcolr_rep colr_rep;            /* target address             */
      } colr;
      struct Pimage_resource {
         Pint      image_resource_id;   /* image resource identifier  */
      } image_resource;
      ...                               /* implementation defined     */
   } data;

} Pbackg_method;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   Pint                type;           /* trav. res. operation type   */
   union Ptrav_res_op_data {
      struct Ptrav_res_clear {
         Pint      res_id;             /* traversal resource id.      */
      } trav_res_clear;
      struct Ptrav_res_copy {
         Pint      src_res_id;         /* source traversal res. id.   */
         Pint      dest_res_id;        /* destination trav. res. id.  */
      } trav_res_copy;
      ...                              /* implementation defined      */
   } data;                             /* trav. res. operation data   */
} Ptrav_res_op;

typedef struct {
   union Pdi_pick_pets {
      struct Pdi_pick_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pdi_pick_pet_r1 {
         Pint      imp_dep;            /* impl. dependent             */
      } pet_r1;  /* For PET 1 */
      ...                              /* implementation defined PET's*/
   } pets;
   union Pdi_pick_measure_data {
      struct Pdi_pick_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Pdi_pick_trigger_data {
      struct Pdi_pick_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined di_pick processes */
   } trigger_data;
   union Pdi_pick_ack_data {
      struct Pdi_pick_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pdi_pick_data;

typedef struct {
   union Pdi_pick3_pets {
      struct Pdi_pick3_pet_other {
         Pint      unused;
      } pet_other;  /* When no echo-specific data is required */
      struct Pdi_pick3_pet_r1 {
         Pint      imp_dep;           /* impl. dependent            */
      } pet_r1;  /* For PET 1 */
      Pint      imp_dep; /* data for impl. defined pets */
   } pets;
   union Pdi_pick3_measure_data {
      struct Pdi_pick3_meas_other {
         Pint      unused;
      } meas_other; /* When no measure-specific data is required */
      Pint      imp_dep; /* data for impl. defined measure processes */
   } measure_data;
   union Pdi_pick3_trigger_data {
      struct Pdi_pick3_trig_other {
         Pint      unused;
      } trig_other; /* When no trigger-specific data is required */
      Pint      imp_dep; /* data for impl. defined di_pick processes */
   } trigger_data;
   union Pdi_pick3_ack_data {
      struct Pdi_pick3_ack_other {
         Pint      unused;
      } ack_other; /* When no ack.-specific data is required */
      Pint      imp_dep; /* data for impl. defined ack. processes */
   } acknowledgement_data;

} Pdi_pick_data3;

typedef struct {
   Pint_list  alpha_srcs;      /* list available alpha sources       */
   Pint_list  transp_modes;    /* list transparency modes supported  */

} Palpha_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pdyn_mod     texture_rep;          /* texture representation          */
} Pdyns_ws_attrs_texture;

typedef struct {
  Pint      max_dims[3];         /* max dimensions mipmap base level   */
  Pboolean  equal_dims_req;      /* equal mipmap dimensions required   */
  Pboolean  power_2_dims_req;    /* power of 2 mipmap dimensions req.  */
} Pmipmap_facs;

typedef enum {
   PPERSP_CORR_NONE = 0,
   PPERSP_CORR_AT_VERTICES = 1,
   PPERSP_CORR_INTERIOR = 2
} Pperspect_corr;

typedef struct {
   Pint_list  avail_pick_types;      /* list available pick types     */
   Pint_list  avail_echo_types;      /* list of available echo types  */
} Ppick_mapping_facs;

typedef enum {
  PRES_HINTS_UTILIZED_NO,
  PRES_HINTS_UTILIZED_YES
} Pres_hints;

typedef struct {
  Pint      image_res_id;        /* texture image resource identifier */
  Pint      rendering_order;     /* rendering order                   */
} Ptexture_binding;

typedef struct {
   Pint           method;          /* composition method              */
   union {
      Pint        unused;          /* used by methods 1, 2, and 4     */
      struct {                     /* used by method 3                */
        Pgcolr    environ;         /* environment colour              */
        Pint      gamma_chan_sel;  /* gamma channel selector          */
      } blend_env;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        Pgcolr      background_colr; /* used by method 5              */
        ...                          /* implementation defined        */
    } data;
} Ptexture_compos;

typedef struct {
  Pint_list  coord_srcs;       /* list avail. coordinate sources     */
  Pint_list  compos_methods;   /* list avail. composition methods    */
  Pint_list  min_methods;      /* list avail. minification methods   */
  Pint_list  mag_methods;      /* list avail. magnification methods  */
  Pint_list  bound_cond;       /* list avail. boundary conditions    */
  Pint_list  clamp_methods;    /* list avail. boundary clamp methods */
  Pint_list  render_orders;    /* list avail. rendering orders       */
  Pint       num_pred_inds;    /* number predefined texture indices  */
} Ptexture_facs;

typedef enum {
  PTUP_UTILIZED_NO = 0,
  PTUP_UTILIZED_YES = 1
} Ptup_util;

typedef struct {
  Pint       max_simul;        /* max number simultaneously          */
                               /* appliable textures                 */
  Pint_list  perspect_corr;    /* list of available perspective      */
                               /* correction methods                 */
  Pint_list  sampling_freqs;   /* list avail. sampling frequencies   */
  Pint_list  opt_hints;        /* list of available resource         */
                               /* optimization hints                 */
  Ptup_util  usage;            /* utilization of texture usage       */
                               /* priorities                         */
  Pint_list  image_res_types;  /* list of image resource types       */
                               /* available for texture mapping      */
} Ptexture_map_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
   unsigned int coord_src:1;
   unsigned int coord_src_data:1;
   unsigned int orientation:1;
   unsigned int compos_method:1;
   unsigned int compos_data:1;
   unsigned int min_method:1;
   unsigned int max_method:1;
   unsigned int bound_conds:1;
   unsigned int clamp_method:1
   unsigned int clamp_data:1
   unsigned int depth_sampl_hint:1
   unsigned int freq_wt_hints:1
   unsigned int image_res_id:1
   unsigned int rendering_order:1
   unsigned int :18
} Ptexture_mask;

typedef struct {
   Pint           coord_src;       /* coordinate source                */
   union Pcoord_src_data {         /* coordinate source data record    */
      Pint        unused;          /* used by coord sources 1 - 5      */
      Pint        coord_ind;       /* used by coord source 6           */
      Pmatrix3    refl_matrix;     /* used by coord sources 7 and 8    */
      ...                          /* implementation defined           */
   } data;
} Pcoord_src;

typedef struct {
  Pcoord_src      coord_src;          /* coordinate source and data  */
  Pmatrix3        orientation;        /* orientation matrix          */
} Ptexture_param;

typedef struct {
   Pint           method;          /* clamp method                     */
   union Pclamp_method_data {
      Pint        unused;          /* used by method 1                 */
      Pgcolr      clamp_colr;      /* used by method 2                 */
      ...                          /* implementation defined           */
   } data;
} Pclamp_method;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {

  Pint            min_method;          /* minification method          */
  Pint            mag_method;          /* magnification method         */
  Pint            bound_conds[3];      /* boundary conditions          */
      Pclamp_methodclamp;              /* boundary clamp method & data*/
  Pfloat          depth_sampl_hint;    /* depth sampling hint          */
  Pfloat          freq_wt_hints[3];    /* frequency weight hints       */

} Ptexture_sampling;

typedef struct {

  Ptexture_param     param;            /* texture parametrization      */
  Ptexture_compos    compos;           /* texture composition          */
  Ptexture_sampling  sampling;         /* texture sampling             */
  Ptexture_binding   binding;          /* texture binding              */

} Ptexture_rep;

typedef struct {

   Pint              opt_hint;          /* optimization hint          */
   Pint_list         usage_priorities;  /* texture usage priorities */

} Ptexture_res_opt_heur;

typedef struct {

   Pint      max_simul_trig,    /* max simul. trigger procs              */
   Pint      max_simul_echo,    /* max simul. echo procs                 */
   Pint      max_simul_ack,     /* max simul. acknowledgement procs      */
   Pint_list meas_proc_ids,     /* available measure process ids         */
   Pint_list trig_proc_ids,     /* available trigger process ids         */
   Pint_list echo_proc_ids,     /* available echo process ids            */
   Pint_list ack_proc_ids       /* available acknowledgement process ids
                                   */

} Pval_facs;

typedef struct {

   Pint   highl_rep;         /* max. # of highlighting table entries  */

} Pws_st_tables_highl;

typedef struct {

   Pint   texture_rep;          /* max. # of texture table entries    */

} Pws_st_tables_texture;"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Page 195

The following text is inserted before the definition of pelem_data:

"

```
typedef struct {

   Pint           num_pred;        /* number of predefined methods   */
   Pint_list      methods;         /* number of highl. methods supp. */
   Pint           max_highl_ind;   /* max highl. table indices       */

} Phighl_facs;

typedef struct {

   Pint                 method;

   union _Phighl_data {
      Pint         unused;
      Pfloat       blink_rate;
      Pint         colr_ind;
      Pgcolr       colr;
      struct _Phighl_blink_colr {
         Pfloat    blink_rate;
         Pgcolr    blink_colr;
      } highl_blink_colr;
      ...          /* implementation defined methods*/
   } highl_data;

} Phighl_method;

typedef enum {

   PTRL_EQUAL = 0,
   PTRL_NOT_EQUAL = 1,
   PTRL_GREATER = 2,
   PTRL_LESS = 3,
   PTRL_GREATER_OR_EQUAL = 4,
   PTRL_LESS_OR_EQUAL = 5,
   PTLG_BITWISE_AND = 6,
   PTLG_BITWISE_OR = 7,
   PTLG_BITWISE_XOR = 8

} Ptest_comp;

typedef enum {

   PFILTER_FAILS = 0,
   PFILTER_PASSES = 1

} Pfilter_op;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
   PCOND_ALL_ENABLED,
   PCOND_ALL_DISABLED,
   PCOND_NOT_ALL_DISABLED,
   PCOND_NOT_ALL_ENABLED,
   PCOND_ALWAYS_PASS,
   PCOND_NEVER_PASS
} Pcond_flags_eval_op;

typedef union {
   unsigned int mask;
   struct _Pcond_flag_set {
      unsigned int flag0:1;
      unsigned int flag1:1;
      unsigned int flag2:1;
      unsigned int flag3:1;
      unsigned int flag4:1;
      unsigned int flag5:1;
      unsigned int flag6:1;
      unsigned int flag7:1;
      unsigned int flag8:1;
      unsigned int flag9:1;
      unsigned int flag10:1;
      unsigned int flag11:1;
      unsigned int flag12:1;
      unsigned int flag13:1;
      unsigned int flag14:1;
      unsigned int flag15:1;
      unsigned int flag16:1;
      unsigned int flag17:1;
      unsigned int flag18:1;
      unsigned int flag19:1;
      unsigned int flag20:1;
      unsigned int flag21:1;
      unsigned int flag22:1;
      unsigned int flag23:1;
      unsigned int flag24:1;
      unsigned int flag25:1;
      unsigned int flag26:1;
      unsigned int flag27:1;
      unsigned int flag28:1;
      unsigned int flag29:1;
      unsigned int flag30:1;
      unsigned int flag31:1;
   } flags;
} Pcond_flag_mask;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
   Pint                 method;          /* test method                  */
   union _Ptest_data {
      struct Ptest_always_fail {
         Pint              unused;
      } test_always_fail;
      struct Ptest_always_succeed {
         Pint              unused;
      } test_always_succeed;
      struct Ptest_extent_3 {
         Plimit3           rect;          /* extent rectangle             */
         Ptest_comp        relation;      /* test relation                */
         Pfloat            threshhold;    /* threshhold value             */
      } test_extent_3;
      struct Ptest_extent {
         Plimit            rect;          /* extent rectangle             */
         Ptest_comp        relation;      /* test relation                */
         Pfloat            threshhold;    /* threshhold value             */
      } test_extent;
      struct Ptest_bounds_3 {
         Ppoint_list_list3 point_lists;   /* test object                  */
         Pclip_cond        cond;          /* test condition               */
      } test_bounds_3;
      struct Ptest_bounds {
         Ppoint_list_list  point_lists;   /* test object                  */
         Pclip_cond        cond;          /* test condition               */
      } test_bounds;
      struct Ptest_nameset {
         Pint              filter_type;   /* type of filter               */
         Pint              filter_id;     /* filter identifier            */
         Pfilter_op        cond;          /* test condition               */
      } test_nameset;
      struct Ptest_appl_int {
         Pint              appl_id;       /* appl. integer selector       */
         Ptest_comp        comparison;    /* test comparison              */
         Pint              data;          /* appl integer data value      */
      } test_appl_int;
      struct Ptest_appl_real {
         Pint              appl_id;       /* appl. real selector          */
         Ptest_comp        comparison;    /* test comparison              */
         Pfloat            data;          /* appl real data value         */
      } test_appl_real;
      struct Ptest_cond_flags {
         Pcond_flags_eval_op  op;      /* condition flags eval. op.       */
      } test_cond_flags;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
      struct Ptest_appl_int_as_logical {
         Pint               appl_id;     /* appl. integer selector   */
         Ptest_comp_logical comparison;  /* test comparison          */
         Pint               data;        /* appl integer data value  */
      } test_appl_int;
      ...                                /* implementation defined   */
   } test_data;       /*                 test data record*/

} Ptest;"
```

Pages 195 and 196

The following text is merged into the text of the definition of pelem_data*:*

```
"
 typedef union {
  /* start of PHIGS element data */
  ...
  /* end of PHIGS element data */
  /* start of PHIGS PLUS element data */
  ...
  /* end of PHIGS PLUS element data */
  /* start of Full PHIGS element data */
  struct Pappl_int {
   Pint   integer_id;    /* application integer identifier          */
   Pint   value;         /* value for application integer           */
  } appl_int;
  struct Pappl_real {
   Pint      real_id;       /* application real identifier          */
   Pfloat    value;         /* value for application real           */
  } appl_real;
  struct Pcircle3 {
      Ppoint3     center_point;      /* center point                 */
      Pfloat      radius;            /* radius                       */
      Pvec3       ref_vecs[2];       /* reference vectors            */
  } circle3;
  struct Pcircle {
   Ppoint      center_point;         /* center point                 */
   Pfloat      radius;               /* radius                       */
  } circle;
  struct Pcircular_arc3 {
   Ppoint3            center_point;  /* center point                 */
   Pfloat             radius;        /* radius                       */
   Pvec3              ref_vecs[2];   /* reference vectors            */
   Pfloat             start;         /* start angle                  */
   Pfloat             end;           /* end angle                    */
  } circular_arc3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
struct Pcircular_arc {
  Ppoint              center_point;   /* center point                */
  Pfloat              radius;         /* radius                      */
  Pfloat              start;          /* start angle                 */
  Pfloat              end;            /* end angle                   */
} circular_arc;
struct Pellipse3 {
  Ppoint3          center_point;      /* center point                */
  Pvec3            major_ref_vec;     /* major axis reference vector */
  Pvec3            minor_ref_vec;     /* minor axis reference vector */
} ellipse3;
struct Pellipse {
  Ppoint           center_point;      /* center point                */
  Pvec             major_ref_vec;     /* major axis reference vector */
  Pvec             minor_ref_vec;     /* minor axis reference vector */
} ellipse;
struct Pelliptical_arc3 {
  Ppoint3          center_point;      /* center point                */
  Pvec3            major_ref_vec;     /* major axis reference vector */
  Pvec3            minor_ref_vec;     /* minor axis reference vector */
  Pfloat           start;             /* start angle                 */
  Pfloat           end;               /* end angle                   */
} elliptical_arc3;
struct Pelliptical_arc {
  Ppoint           center_point;      /* center point                */
  Pvec             major_ref_vec;     /* major axis reference vector */
  Pvec             minor_ref_vec;     /* minor axis reference vector */
  Pfloat           start;             /* start angle                 */
  Pfloat           end;               /* end angle                   */
} elliptical_arc;
struct Pfill_circle3 {
  Ppoint3      center_point;          /* center point                */
  Pfloat       radius;                /* radius                      */
  Pvec3        ref_vecs[2];           /* reference vectors           */
} fill_circle3;
struct Pfill_circle {
  Ppoint       center_point;          /* center point                */
  Pfloat       radius;                /* radius                      */
} fill_circle;
struct Pcircular_arc_close3 {
  Ppoint3             center_point;   /* center point                */
  Pfloat              radius;         /* radius                      */
  Pvec3               ref_vecs[2];    /* reference vectors           */
  Pfloat              start;          /* start angle                 */
  Pfloat              end;            /* end angle                   */
  Pclosure            type;           /* closure type                */
} circular_arc_close3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
struct Pcircular_arc_close {
  Ppoint              center_point;  /* center point                */
  Pfloat              radius;        /* radius                      */
  Pfloat              start;         /* start angle                 */
  Pfloat              end;           /* end angle                   */
  Pclosure            type;          /* closure type                */
} circular_arc_close;

struct Pfill_ellipse3 {
  Ppoint3         center_point;      /* center point                */
  Pvec3           major_ref_vec;     /* major axis reference vector */
  Pvec3           minor_ref_vec;     /* minor axis reference vector */
} fill_ellipse3;

struct Pfill_ellipse {
  Ppoint          center_point;      /* center point                */
  Pvec            major_ref_vec;     /* major axis reference vector */
  Pvec            minor_ref_vec;     /* minor axis reference vector */
} fill_ellipse;

struct Pelliptical_arc_close3 {
  Ppoint3         center_point;      /* center point                */
  Pvec3           major_ref_vec;     /* major axis reference vector */
  Pvec3           minor_ref_vec;     /* minor axis reference vector */
  Pfloat          start;             /* start angle                 */
  Pfloat          end;               /* end angle                   */
  Pclosure        type;              /* closure type                */
} elliptical_arc_close3;

struct Pelliptical_arc_close {
  Ppoint          center_point;      /* center point                */
  Pvec            major_ref_vec;     /* major axis reference vector */
  Pvec            minor_ref_vec;     /* minor axis reference vector */
  Pfloat          start;             /* start angle                 */
  Pfloat          end;               /* end angle                   */
  Pclosure        type;              /* closure type                */
} elliptical_arc_close;

Plinetype_adapt  linetype_adapt;     /* linetype adaptability       */
Plinetype_cont   linetype_cont;      /* linetype continuity         */
Phighl_method    highl_method;       /* highlighting method         */
Pint_list        active_textures;      /* active textures            */
Pint_list        back_active_textures; /* back active textures       */
Pperspect_corr   perspect_corr;   /* textr perspective correction   */

struct Ptexture_res_opt_heur {
   Pint          opt_hint;          /* optimization hint           */
   Pint_list     usage_priorities; /* texture usage priorities     */
} texture_res_opt_heur;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
   Pint_list        alpha_src_sel;   /* alpha source selector          */
   struct Pcond_exec_struct {
    Pint            struct_id;          /* structure identifier        */
    Ptest           test;               /* condition test              */
   } cond_exec_struct;
   struct Pcond_inst_struct {
    Pint            struct_id;          /* structure identifier        */
    Ptest           test;               /* condition test              */
   } cond_inst_struct;
   Ptest            cond_return;        /* condition test              */
   struct Pcond_skip_elems {
    Pint            skip_count;         /* number of elements to skip  */
    Ptest           test;               /* condition test              */
   } cond_skip_elems;
   struct Pcond_skip_to_label {
    Pint            label;              /* label to which to skip      */
    Ptest           test;               /* condition test              */
   } cond_skip_to_label;
   /* end of Full PHIGS element data */

  } Pelem_data;"
```

Page 247

A.3 External functions

The following text is appended:

```
"/* WORKSTATION TYPE CREATE */

void pws_type_create (

   Pint                    ws_type,        /* workstation type          */
   const Pconfig_setting   *settings_list,/* array of config. settings*/
   Pint                    *new_ws_type    /* OUT new workstation type  */

);

/* WORKSTATION TYPE SET */

void pws_type_set (

   Pint                    ws_type,        /* workstation type          */
   const Pconfig_setting   *settings_list /* array of config. settings*/

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* WORKSTATION TYPE GET */

void pws_type_get (

   Pint           ws_type,         /* workstation type                */
   const char     *setting_name,   /* configuration setting name      */
   Pstore         store,           /* store object for returned value */
   Pconfig_value  **setting_value  /* OUT configuration setting value */

);

/* WORKSTATION TYPE DESTROY */

void pws_type_destroy (

   Pint        ws_type                  /* workstation type         */

);

/* REDRAW ALL STRUCTURES ON TARGET */

void predraw_all_structs_on_targ (

   Pint               ws_id,           /* workstation identifier      */
   const Ptarg_addr   *targ_addr,      /* target address              */
   Pctrl_flag         ctrl_flag        /* control flag                */

);

/* REDRAW ALL STRUCTURES FROM POSTING GROUP */

void predraw_all_structs_from_grp (

   Pint               ws_id,           /* workstation identifier      */
   Pint               grp_id,          /* posting group               */
   Pbackg_redisplay   redisplay_flag   /* background redisplay flag   */

);

/* UPDATE TARGET */

void pupd_targ (

   Pint               ws_id,           /* workstation identifier      */
   const Ptarg_addr   *targ_addr,      /* target address              */
   Pregen_flag        regen_flag       /* update regeneration flag    */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* DEFINE POSTING GROUP */

void pdefine_post_grp (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 grp_id,         /* posting group identifier    */
   Pint                 ref_grp_id,     /* reference posting group id  */
   Prel_pri             rel_pri,        /* relative priority           */
   Pint                 def_view_ind,   /* default view index          */
   Pbackg_style         backg_style,    /* posting group backg. style  */
   const Pbackg_method  *backg_method,  /* posting group backg. method */
   Pborder_indic        border_indic,   /* posting group border indic. */
   Pint                 border_ind      /* posting group border index  */

);

/* UNDEFINE POSTING GROUP */

void pundefine_post_grp (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id           /* posting group identifier    */

);

/* SET POSTING GROUP STATUS */

void pset_post_grp_status (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id,          /* posting group identifier    */
   Ppost_grp_status    post_grp_status  /* posting group status        */

);

/* SET POSTING GROUP PRIORITY */

void pset_post_grp_priority (

   Pint                ws_id,           /* workstation identifier      */
   Pint                grp_id,          /* posting group identifier    */
   Pint                ref_grp_id,      /* reference posting group id  */
   Prel_pri            rel_pri          /* relative priority           */

);

/* SET POSTING GROUP BACKGROUND STYLE */

void pset_post_grp_backg_style (

   Pint             ws_id,           /* workstation identifier         */
   Pint             grp_id,          /* posting group identifier       */
   Pbackg_style     backg_style      /* posting group background style */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET POSTING GROUP BACKGROUND METHOD */

void pset_post_grp_backg_method (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 grp_id,         /* posting group identifier    */
   const Pbackg_method  *backg_method   /* posting group background    */
                                        /* method                      */

);

/* SET POSTING GROUP BORDER INDICATOR */

void pset_post_grp_border_indicator (

   Pint           ws_id,            /* workstation identifier         */
   Pint           grp_id,           /* posting group identifier       */
   Pborder_indic  border_indic      /* posting group border indicator */

);

/* SET POSTING GROUP BORDER INDEX */

void pset_post_grp_border_ind (

   Pint       ws_id,              /* workstation identifier          */
   Pint       grp_id,             /* posting group identifier        */
   Pint       border_ind          /* posting group border index      */

);

/* ASSOCIATE IMAGE RESOURCE */

void passoc_image_res (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 image_res_id,   /* image resource identifier   */
   const Pint_size      *dims,          /* m,n dimensions              */
   const Pimage_res     *image_res,     /* image resource              */
   Paccess_flag         *access_flag    /* OUT access flag             */

);

/* DISASSOCIATE IMAGE RESOURCE */

void pdisassoc_image_res (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 image_res_id    /* image resource identifier   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET DEVICE COORDINATE CLIP REGIONS 3 */

void pset_dc_clip_regions3 (

   Pint                 ws_id,          /* workstation identifier      */
   const Plimit3_list   *limit_list     /* list of clip regions        */

);

/* SET DEVICE COORDINATE CLIP REGIONS */

void pset_dc_clip_regions (

   Pint                ws_id,           /* workstation identifier      */
   const Plimit_list   *limit_list      /* list of clip regions        */

);

/* SET TARGET MANIPULATION MODE */

void pset_targ_manip_mode (

   Pint             ws_id,              /* workstation identifier      */
   Pint             targ_manip_mode     /* target manipulation mode    */

);

/* SET TARGET DISPOSITION */

void pset_targ_dispos (

  Pint                  ws_id,         /* workstation identifier      */
  const Ptarg_op_list   *targ_ops      /* target operations list      */

);

/* SET DISPLAY TARGET */

void pset_disp_targ (

   Pint                ws_id,           /* workstation identifier      */
   const Ptarg_addr    *targ_addr       /* target address              */

);

/* SET RENDERING TARGET */

void pset_rend_targ (

   Pint                ws_id,           /* workstation identifier      */
   const Ptarg_addr    *targ_addr       /* target address              */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* CLEAR TARGET */

void pclear_targ (

   Pint               ws_id,          /* workstation identifier      */
   const Ptarg_addr   *targ_addr      /* target address              */

);

/* COPY TARGET */

void pcopy_targ (

   Pint               ws_id,          /* workstation identifier      */
   const Ptarg_addr   *src_targ_addr, /* source target address       */
   const Ptarg_addr   *dest_targ_addr /* destination target address  */

);

/* CREATE TARGET */

void pcreate_targ (

   Pint               ws_id,          /* workstation identifier      */
   const Ptarg_addr   *targ_addr      /* target address              */

);

/* DESTROY TARGET */

void pdestroy_targ (

   Pint               ws_id,          /* workstation identifier      */
   const Ptarg_addr   *targ_addr      /* target address              */

);

/* SET STATE OF VISUAL REPRESENTATION */

void pset_st_visual_rep (

   Pint            ws_id,         /* workstation identifier          */
   Pvisual_st      visual_st      /* state of visual representation  */

);

/* SET TARGET STATE OF VISUAL REPRESENTATION */

void pset_targ_st_visual_rep (

   Pint               ws_id,        /* workstation identifier         */
   const Ptarg_addr   *targ_addr,   /* target address                 */
   Pvisual_st         visual_st     /* state of visual representation */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* ASSOCIATE TRAVERSAL RESOURCE */

void passoc_trav_res (

   Pint                ws_id,        /* workstation identifier          */
   Pint                res_id,       /* traversal resource identifier   */
   const Ptarg_addr    *targ_addr    /* source target address           */

);

/* DISASSOCIATE TRAVERSAL RESOURCE */

void pdisassoc_trav_res (

   Pint                ws_id,        /* workstation identifier          */
   Pint                res_id        /* traversal resource identifier   */

);

/* MANIPULATE TRAVERSAL RESOURCE */

void pmanip_trav_res (

  Pint                  ws_id,        /* workstation identifier         */
  Pint                  res_id,       /* traversal resource identifier  */
  const Ptrav_res_op    *manip_data   /* traversal resource operation   */

);

/* RESET ALL TRAVERSAL RESOURCES */

void preset_all_trav_res (

  Pint                  ws_id          /* workstation identifier       */

);

/* RETRIEVE WINDOW SYSTEM COLOUR */

void pret_window_system_colr (

  Pint           ws_id,           /* workstation identifier            */
  const Pgcolr   *colr,           /* colour sought                     */
  Pstore         store,           /* store object for returned value   */
  Pdata          **winsys_colr    /* OUT window system colour          */

);

/* SET APPLICATION INTEGER */

void pset_appl_int (

   Pint    integer_id,    /* application integer identifier           */
   Pint    value          /* value of application integer             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET APPLICATION REAL */

void pset_appl_real (

   Pint    real_id,       /* application real identifier          */
   Pfloat  value          /* value of application real            */

);

/* CIRCLE 3 */

void pcircle3 (

   const Ppoint3      *center_point,   /* center point             */
   Pfloat             radius,         /* radius                   */
   const Pvec3        ref_vecs[2]     /* reference vectors        */

);

/* CIRCLE */

void pcircle (

   const Ppoint       *center_point,   /* center point             */
   Pfloat             radius          /* radius                   */

);

/* CIRCULAR ARC 3 */

void pcircular_arc3 (

   const Ppoint3      *center_point,   /* center point             */
   Pfloat             radius,         /* radius                   */
   const Pvec3        ref_vecs[2],    /* reference vectors        */
   Pfloat             start,          /* start angle              */
   Pfloat             end             /* end angle                */

);

/* CIRCULAR ARC */

void pcircular_arc (

   const Ppoint       *center_point,   /* center point             */
   Pfloat             radius,         /* radius                   */
   Pfloat             start,          /* start angle              */
   Pfloat             end             /* end angle                */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* ELLIPSE 3 */

void pellipse3 (

   const Ppoint3    *center_point,     /* center point                */
   const Pvec3      *major_ref_vec,   /* major axis reference vector */
   const Pvec3      *minor_ref_vec     /* minor axis reference vector */

);

/* ELLIPSE */

void pellipse (

   const Ppoint     *center_point,     /* center point                */
   const Pvec       *major_ref_vec,   /* major axis reference vector */
   const Pvec       *minor_ref_vec     /* minor axis reference vector */

);

/* ELLIPTICAL ARC 3 */

void pelliptical_arc3 (

   const Ppoint3    *center_point,     /* center point                */
   const Pvec3      *major_ref_vec,   /* major axis reference vector */
   const Pvec3      *minor_ref_vec,   /* minor axis reference vector */
   Pfloat           start,            /* start angle                 */
   Pfloat           end               /* end angle                   */

);

/* ELLIPTICAL ARC */

void pelliptical_arc (

   const Ppoint     *center_point,     /* center point                */
   const Pvec       *major_ref_vec,   /* major axis reference vector */
   const Pvec       *minor_ref_vec,   /* minor axis reference vector */
   Pfloat           start,            /* start angle                 */
   Pfloat           end               /* end angle                   */

);

/* FILL CIRCLE 3 */

void pfill_circle3 (

   const Ppoint3       *center_point, /* center point                */
   Pfloat              radius,        /* radius                      */
   const Pvec3         ref_vecs[2]    /* reference vectors           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* FILL CIRCLE */

void pfill_circle (

   const Ppoint        *center_point,   /* center point                */
   Pfloat              radius          /* radius                      */

);

/* CIRCULAR ARC CLOSE 3 */

void pcircular_arc_close3 (

   const Ppoint3       *center_point,   /* center point                */
   Pfloat              radius,         /* radius                      */
   const Pvec3         ref_vecs[2],    /* reference vectors           */
   Pfloat              start,          /* start angle                 */
   Pfloat              end,            /* end angle                   */
   Pclosure            type            /* closure type                */

);

/* CIRCULAR ARC CLOSE */

void pcircular_arc_close (

   const Ppoint        *center_point,   /* center point                */
   Pfloat              radius,         /* radius                      */
   Pfloat              start,          /* start angle                 */
   Pfloat              end,            /* end angle                   */
   Pclosure            type            /* closure type                */

);

/* FILL ELLIPSE 3 */

void pfill_ellipse3 (

   const Ppoint3    *center_point,     /* center point                */
   const Pvec3      *major_ref_vec,   /* major axis reference vector */
   const Pvec3      *minor_ref_vec    /* minor axis reference vector */

);

/* FILL ELLIPSE */

void pfill_ellipse (

   const Ppoint     *center_point,     /* center point                */
   const Pvec       *major_ref_vec,   /* major axis reference vector */
   const Pvec       *minor_ref_vec    /* minor axis reference vector */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* ELLIPTICAL ARC CLOSE 3 */

void pelliptical_arc_close3 (

   const Ppoint3    *center_point,     /* center point                */
   const Pvec3      *major_ref_vec,   /* major axis reference vector */
   const Pvec3      *minor_ref_vec,   /* minor axis reference vector */
   Pfloat           start,            /* start angle                 */
   Pfloat           end,              /* end angle                   */
   Pclosure         type              /* closure type                */

);

/* ELLIPTICAL ARC CLOSE */

void pelliptical_arc_close (

   const Ppoint     *center_point,     /* center point                */
   const Pvec       *major_ref_vec,   /* major axis reference vector */
   const Pvec       *minor_ref_vec,   /* minor axis reference vector */
   Pfloat           start,            /* start angle                 */
   Pfloat           end,              /* end angle                   */
   Pclosure         type              /* closure type                */

);

/* SET HIGHLIGHTING INDEX */

void pset_highl_ind (

   Pint       highl_ind               /* highlighting index          */

);

/* SET LINETYPE ADAPTABILITY */

void pset_linetype_adapt (

   Plinetype_adapt    adaptability    /* linetype adaptability       */

);

/* SET LINETYPE CONTINUITY */

void pset_linetype_cont (

   Plinetype_cont     continuity      /* linetype continuity         */

);

/* SET LINETYPE OFFSET */

void pset_linetype_offset (

   Pfloat             offset          /* linetype offset             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET HIGHLIGHTING METHOD */

void pset_highl_method (

   const Phighl_method  *method          /* highlighting method          */

);

/* SET CONDITION FLAGS */

void pset_cond_flags (

   const Pcond_flags   *enable_mask,     /* condition flag enable mask  */
   const Pcond_flags   *disable_mask     /* condition flag disable mask */

);

/* SET CONDITION FLAGS FROM TESTS */

void pset_cond_flags_from_tests (

   const Pcond_test_list   *tests        /* list of condition flag tests
                                            */

);

/* SET ACTIVE TEXTURES */

void pset_active_textures (

   const Pint_list  *active_textures     /* list of active textures      */

);

/* SET BACK ACTIVE TEXTURES */

void pset_back_active_textures (

   const Pint_list  *back_active_textures /* list of active textures   */

);

/* SET TEXTURE PERSPECTIVE CORRECTION */

void pset_texture_perspect_corr (

   Pperspect_corr   perspect_corr     /* texture perspective correction */

);

/* SET TEXTURE SAMPLING FREQUENCY */

void pset_texture_sampling_freq (

   Pint           sampling_freq          /* texture sampling frequency  */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET TEXTURE RESOURCE OPTIMIZATION HEURISTICS */

void pset_texture_res_opt_heur (

   Pint             opt_hint,           /* optimization hint           */
   const Pint_list  *usage_priorities   /* texture usage priorities    */

);

/* SET TRANSPARENCY */

void pset_transparency (

   Pfloat        transparency           /* transparency                */

);

/* SET BACK TRANSPARENCY */

void pset_back_transparency (

  Pfloat         back_transparency      /* back transparency           */

);

/* SET ALPHA SOURCE SELECTOR */

void pset_alpha_src_sel (

   const Pint_list  *alpha_src_sel      /* alpha source selector list  */

);

/* SET ALPHA DATA SELECTION INDEX */

void pset_alpha_data_sel_ind (

   Pint       alpha_data_sel_ind        /* alpha data selection index  */

);

/* DEFINE LINETYPE */

void pdefine_linetype (

   Pint               linetype,         /* linetype                    */
   Pfloat             repeat_length,    /* repeat length               */
   const Pfloat_list  *segment_lengths  /* segment length list         */

);

/* DEFINE MARKER TYPE */

void pdefine_marker_type (

   Pint               marker_type,      /* marker type                 */
   const Pmarker_desc *marker           /* marker descriptor           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET EXTENDED PATTERN REPRESENTATION */

void pset_ext_pat_rep (

   Pint              ws_id,        /* workstation identifier          */
   Pint              pat_ind,      /* pattern bundle index            */
   const Ppattern    *pattern      /* extended pattern representation */

);

/* SET HIGHLIGHTING REPRESENTATION */

void pset_highl_rep (

   Pint                 ws_id,          /* workstation identifier      */
   Pint                 highl_ind,      /* highlighting bundle index   */
   const Phighl_method  *method         /* highlighting method         */

);

/* SET TEXTURE  REPRESENTATION */

void pset_texture_rep (

   Pint                    ws_id,     /* workstation identifier        */
   Pint                    index,     /* texture index                 */
   const Ptexture_rep      *rep       /* texture representation        */

);

/* SET TEXTURE PARAMETRIZATION */

void pset_texture_param (

   Pint                  ws_id,   /* workstation identifier           */
   Pint                  index,   /* texture index                    */
   const Ptexture_param  *param   /* texture parametrization          */

);

/* SET TEXTURE COMPOSITION */

void pset_texture_composition (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    index,       /* texture index               */
   const Ptexture_compos   *compos      /* texture composition         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET TEXTURE SAMPLING */

void pset_texture_sampling (

   Pint                     ws_id,      /* workstation identifier      */
   Pint                     index,      /* texture index               */
   const Ptexture_sampling  *sampling   /* texture sampling            */

);

/* SET TEXTURE BINDING */

void pset_texture_binding (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    index,       /* texture index               */
   const Ptexture_binding  *binding     /* texture binding             */

);

/* SET TRANSPARENCY MODE */

void pset_transparency_mode (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    mode         /* transparency mode           */

);

/* SET TRANSPARENCY THRESHOLDS */

void pset_transparency_thresholds (

   Pint              ws_id,         /* workstation identifier          */
   const Pfloat_list *thresholds    /* list of transparency thresholds */

);

/* SET APPLICATION FILTER */

void pset_appl_filter (

   Pint           ws_id,            /* workstation identifier      */
   Pint           filter_selector,  /* application filter selector */
   const Pfilter  *filter           /* application filter           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* CREATE MIPMAP TEXTURE */

void pcreate_mipmap_texture (

   Pint     ws_id,                  /* workstation identifier          */
   Pint     src_image_res_id,       /* source image resource id        */
   Pint     mipmap_image_res_id,    /* mipmap image resource id        */
   Pint     max_levels,             /* max mipmap levels               */
   Pint     creation_heuristic      /* mipmap creation heuristic       */

);

/* SET VIEW REFERENCE POINT 3 */

void pset_view_ref_point3 (

   Pint            ws_id,           /* workstation id                  */
   Pint            view_ind,        /* view index                      */
   const Ppoint3   *view_ref_point  /* view reference point            */

);

/* SET VIEW REFERENCE POINT */

void pset_view_ref_point (

   Pint            ws_id,           /* workstation id                  */
   Pint            view_ind,        /* view index                      */
   const Ppoint    *view_ref_point  /* view reference point            */

);

/* SET VIEW PLANE NORMAL */

void pset_view_plane_norm (

   Pint              ws_id,         /* workstation id                  */
   Pint              view_ind,      /* view index                      */
   const Pvec3       *view_norm_vec /* view normal vector              */

);

/* SET VIEW UP VECTOR 3 */

void pset_view_up_vec3 (

   Pint              ws_id,         /* workstation id                  */
   Pint              view_ind,      /* view index                      */
   const Pvec3       *view_up_vec   /* view up vector                  */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET VIEW UP VECTOR */

void pset_view_up_vec (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Pvec        *view_up_vec    /* view up vector              */

);

/* SET VIEW WINDOW LIMITS */

void pset_view_win_limits (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Plimit      *win            /* view window limits          */

);

/* SET PROJECTION VIEWPORT 3 */

void pset_proj_vp3 (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Plimit3     *proj_vp        /* projection viewport limits  */

);

/* SET PROJECTION VIEWPORT */

void pset_proj_vp (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   const Plimit      *proj_vp        /* projection viewport limits  */

);

/* SET PROJECTION TYPE */

void pset_proj_type (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pproj_type        proj_type       /* projection type             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET PROJECTION REFERENCE POINT */

void pset_proj_ref_point (

   Pint              ws_id,              /* workstation id                */
   Pint              view_ind,           /* view index                    */
       const Ppoint3*proj_ref_point      /* projection  referernce point*/

);

/* SET VIEW PLANE DISTANCE */

void pset_view_plane_dist (

   Pint                ws_id,            /* workstation id                */
   Pint                view_ind,         /* view index                    */
   Pfloat              view_plane        /* view plane distance           */

);

/* SET FRONT PLANE DISTANCE */

void pset_front_plane_dist (

   Pint                ws_id,            /* workstation id                */
   Pint                view_ind,         /* view index                    */
   Pfloat              front_plane       /* front plane distance          */

);

/* SET BACK PLANE DISTANCE */

void pset_back_plane_dist (

   Pint                ws_id,            /* workstation id                */
   Pint                view_ind,         /* view index                    */
   Pfloat              back_plane        /* back plane distance           */

);

/* SET X-Y CLIPPING INDICATOR */

void pset_xy_clip_indicator (

   Pint                ws_id,            /* workstation id                */
   Pint                view_ind,         /* view index                    */
   Pclip_ind           xy_clip           /* xy clipping indicator         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET FRONT CLIPPING INDICATOR */

void pset_front_clip_indicator (

   Pint                ws_id,           /* workstation id                */
   Pint                view_ind,        /* view index                    */
   Pclip_ind           front_clip       /* front clipping indicator      */

);

/* SET BACK CLIPPING INDICATOR */

void pset_back_clip_indicator (

   Pint                ws_id,           /* workstation id                */
   Pint                view_ind,        /* view index                    */
   Pclip_ind           back_clip        /* back clipping indicator       */

);

/* UPDATE VIEW REPRESENTATION */

void pupd_view_rep (

   Pint             ws_id,              /* workstation id                */
   Pint             view_ind,           /* view index                    */
   Pint             view_upd_method     /* view update method            */

);

/* MAP DEVICE COORDINATES TO WINDOW SYSTEM COORDINATES */

void pmap_dc_to_wsc (

   Pint              ws_id,   /* workstation id                          */
   const Ppoint_list *dc,     /* device coordinate points                */
   Ppoint_list       *wsc     /* OUT window system coordinate points     */

);

/* MAP WINDOW SYSTEM COORDINATES TO DEVICE COORDINATES */

void pmap_wsc_to_dc (

   Pint              ws_id,       /* workstation id                      */
   const Ppoint_list *wsc,        /* window system coordinate points     */
   Ppoint_list       *dc          /* OUT device coordinate points        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* MAP DEVICE COORDINATES TO WORLD COORDINATES */

void pmap_dc_to_wc (

   Pint              ws_id,       /* workstation id                  */
   const Ppoint_list *dc,         /* device coordinate points        */
   Ppoint_list       *wc          /* OUT world coordinate points     */

);

/* OPEN DIRECT INTERPRETATION STRUCTURE */

void popen_di_struct (

   void

);

/* CLOSE DIRECT INTERPRETATION STRUCTURE */

void pclose_di_struct (

   void

);

/* INSTANCE STRUCTURE */

void pinst_struct (

   Pint         struct_id              /* structure identifier       */

);

/* CONDITIONAL EXECUTE STRUCTURE */

void pcond_exec_struct (

   Pint              struct_id,       /* structure identifier       */
   const Ptest       *test            /* condition test             */

);

/* CONDITIONAL INSTANCE STRUCTURE */

void pcond_inst_struct (

   Pint              struct_id,       /* structure identifier       */
   const Ptest       *test            /* condition test             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* CONDITIONAL RETURN */

void pcond_return (

   const Ptest        *test            /* condition test              */

);

/* CONDITIONAL SKIP ELEMENTS */

void pcond_skip_elements (

   Pint          skip_count,           /* number of elements to skip  */
   const Ptest   *test                 /* condition test              */

);

/* CONDITIONAL SKIP TO LABEL */

void pcond_skip_to_label (

   Pint          label,                /* label to which to skip      */
   const Ptest   *test                 /* condition test              */

);

/* PUSH STATE */

void ppush_st (

   Pint              struct_id,        /* structure identifier        */
   Pref_type         ref_type          /* reference type              */

);

/* POP STATE */

void ppop_st (

   void

);

/* COPY ELEMENT FROM STRUCTURE */

void pcopy_elem_struct (

   Pint          struct_id,            /* structure identifier        */
   Pint          elem_num              /* element number              */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* COPY ELEMENT RANGE FROM STRUCTURE */

void pcopy_elem_range_struct (

   Pint          struct_id,           /* structure identifier       */
   Pint          elem_ptr_1_value,    /* element position 1         */
   Pint          elem_ptr_2_value     /* element position 2         */

);

/* COPY ELEMENTS BETWEEN LABELS */

void pcopy_elems_between_labels_struct (

   Pint          struct_id,           /* structure identifier       */
   Pint          label1_id,           /* label identifier 1         */
   Pint          label2_id            /* label identifier 2         */

);

/* MOVE ELEMENT */

void pmove_elem (

   Pint          struct_id,           /* structure identifier       */
   Pint          elem_num             /* element number             */

);

/* MOVE ELEMENT RANGE */

void pmove_elem_range (

   Pint          struct_id,           /* structure identifier       */
   Pint          elem_ptr_1_value,    /* element position 1         */
   Pint          elem_ptr_2_value     /* element position 2         */

);

/* MOVE ELEMENTS BETWEEN LABELS */

void pmove_elems_between_labels (

   Pint          struct_id,           /* structure identifier       */
   Pint          label1_id,           /* label identifier 1         */
   Pint          label2_id            /* label identifier 2         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET WATCH ON ELEMENT RANGE */

void pset_watch_on_elem_range (

   Pint      struct_id,              /* watched structure identifier*/
   Pint      elem_ptr1_value,        /* element pointer 1 value     */
   Pint      elem_ptr2_value         /* element pointer 2 value     */

);

/* END WATCH ON ELEMENT RANGE */

void pend_watch_on_elem_range (

   void

);

/* POST  STRUCTURE TO GROUP */

void ppost_struct_to_grp (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* posting group identifier    */
   Pint              struct_id,      /* structure identifier        */
   Pfloat            pri             /* priority                    */

);

/* UNPOST  STRUCTURE FROM GROUP */

void punpost_struct_from_grp (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id,         /* posting group identifier    */
   Pint              struct_id       /* structure identifier        */

);

/* UNPOST ALL STRUCTURES FROM GROUP */

void punpost_all_structs_from_grp (

   Pint              ws_id,          /* workstation identifier      */
   Pint              grp_id          /* posting group identifier    */

);

/* POST DIRECT INTERPRETATION STRUCTURE */

void ppost_di_struct (

   Pint              ws_id,          /* workstation identifier      */
   Pint              struct_id       /* structure identifier        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* UNPOST DIRECT INTERPRETATION STRUCTURE */

void punpost_di_struct (

   Pint              ws_id           /* workstation identifier     */

);

/* RENEW DIRECT INTERPRETATION STATE */

void prenew_di_state (

   Pint              ws_id,          /* workstation identifier     */
   Pint              struct_id,      /* structure identifier       */
   Ptrav_type        trav_type       /* traversal type             */

);

/* SET DIRECT INTERPRETATION MODE */

void pset_di_mode (

   Pint              ws_id,          /* workstation identifier     */
   Ptrav_type        trav_type,      /* traversal type             */
   Pdi_mode          mode            /* direct interpretation mode */

);

/* MARK MULTI-PASS START */

void pmark_multi_pass_start (

   Pint              ws_id           /* workstation identifier     */

);

/* MARK MULTI-PASS COMPLETION */

void pmark_multi_pass_compl (

   Pint              ws_id           /* workstation identifier     */

);

/* MARK PASS START */

void pmark_pass_start (

   Pint              ws_id           /* workstation identifier     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* MARK PASS COMPLETION */

void pmark_pass_compl (

   Pint                ws_id            /* workstation identifier      */

);

/* RETRIEVE NUMBER OF PASSES REQUIRED */

void pret_num_passes_req (

   Pint          ws_id,              /* workstation identifier         */
   Ptrav_type    trav_type,          /* traversal type                 */
   Pint          *num_passes_req,    /* OUT picture passes required    */
   Pboolean      *delineation_req    /* OUT pass delineations required */

);

/* SET SET PICK FILTER */

void pset_set_pick_filter (

   Pint                ws_id,           /* workstation identifier      */
   Pint                dev,             /* set dev. number             */
   const Pfilter       *filter          /* echo switch                 */

);

/* SET COMPOSITE PICK FILTER */

void pset_composite_pick_filter (

   Pint                ws_id,           /* workstation identifier      */
   Pint                dev,             /* composite dev. number       */
   const Pfilter       *filter          /* echo switch                 */

);

/* SET DIRECT INTERPRETATION PICK FILTER */

void pset_di_pick_filter (

   Pint                ws_id,           /* workstation identifier      */
   const Pfilter       *filter          /* pick filter                 */

);

/* SET DIRECT INTERPRETATION PICK CORRELATION POINT 3 */

void pset_di_pick_corr_point3 (

   Pint                ws_id,           /* workstation identifier      */
   const Ppoint3       *point           /* pick correlation point      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET DIRECT INTERPRETATION PICK CORRELATION POINT */

void pset_di_pick_corr_point (

   Pint               ws_id,          /* workstation identifier      */
   const Ppoint       *point          /* pick correlation point      */

);

/* DEFINE LOCATOR */

void pdefine_locator (

   Pint                    ws_id,            /* workstation id       */
   Pint                    dev,              /* locator dev. number  */
   Pint                    measure_proc_id,  /* locator measure proc.*/
   const Ptrig_process_list *triggers,       /* list of trigger proc.*/
   const Pecho_process_list *echos,          /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.   */

);

/* DEFINE STROKE */

void pdefine_stroke (

   Pint                    ws_id,            /* workstation id       */
   Pint                    dev,              /* stroke dev. number   */
   Pint                    measure_proc_id,  /* stroke measure proc. */
   const Ptrig_process_list *triggers,       /* list of trigger proc.*/
   const Pecho_process_list *echos,          /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.   */

);

/* DEFINE VALUATOR */

void pdefine_valuator (

   Pint                    ws_id,            /* workstation id.      */
   Pint                    dev,              /* valuator dev. number */
   Pint                    measure_proc_id,  /* valuator measure proc*/
   const Ptrig_process_list *triggers,       /* list of trigger proc.*/
   const Pecho_process_list *echos,          /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* DEFINE CHOICE */

void pdefine_choice (

   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* choice dev. number   */
   Pint                      measure_proc_id,  /* choice measure proc. */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);

/* DEFINE PICK */

void pdefine_pick (

   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* pick dev. number     */
   Pint                      measure_proc_id,  /* pick measure proc.   */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);

/* DEFINE STRING */

void pdefine_string (

   Pint                      ws_id,            /* workstation id.      */
   Pint                      dev,              /* string dev. number   */
   Pint                      measure_proc_id,  /* string measure proc. */
   const Ptrig_process_list *triggers,         /* list of trigger proc.*/
   const Pecho_process_list *echos,            /* list of echo proc.   */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);

/* CREATE SET MEASURE */

void pcreate_set_measure (

  Pint                 ws_id,               /* workstation identifier   */
  const Pmeas_process  *base_measure_proc,  /* measure process for set  */
  Pint                 max_measures,        /* maximum measures in set  */
  Pint                 *measure_proc_id     /* OUT set measure process  */
                                            /* identifier               */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* DEFINE SET */

void pdefine_set (

   Pint                      ws_id,             /* workstation id.       */
   Pint                      dev,               /* set dev. number       */
   Pint                      measure_proc_id,   /* set measure proc.     */
   const Ptrig_process_list *triggers,          /* list of trigger proc.*/
   const Pecho_process_list *echos,             /* list of echo proc.    */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);

/* CREATE COMPOSITE MEASURE */

void pcreate_composite_measure (

   Pint                      ws_id,           /* workstation identifier*/
   const Pmeas_process_list *measure_procs,   /* list of measure proc. */
   Pint                     *measure_proc_id /* OUT composite measure */
                                             /* process identifier    */

);

/* DEFINE COMPOSITE */

void pdefine_composite (

   Pint                      ws_id,             /* workstation id.       */
   Pint                      dev,               /* composite dev. number*/
   Pint                      measure_proc_id,   /* composite meas. proc.*/
   const Ptrig_process_list *triggers,          /* list of trigger proc.*/
   const Pecho_process_list *echos,             /* list of echo proc.    */
   const Pack_process_list  *acknowledgements /* list of ack. proc.    */

);

/* UNDEFINE LOCATOR */

void pundefine_locator (

  Pint              ws_id,          /* workstation identifier         */
  Pint              dev             /* locator dev. number            */

);

/* UNDEFINE STROKE */

void pundefine_stroke (

  Pint              ws_id,          /* workstation identifier         */
  Pint              dev             /* stroke dev. number             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* UNDEFINE VALUATOR */

void pundefine_valuator (

  Pint              ws_id,          /* workstation identifier          */
  Pint              dev             /* locator dev. number             */

);

/* UNDEFINE CHOICE */

void pundefine_choice (

  Pint              ws_id,          /* workstation identifier          */
  Pint              dev             /* choice dev. number              */

);

/* UNDEFINE PICK */

void pundefine_pick (

  Pint              ws_id,          /* workstation identifier          */
  Pint              dev             /* pick dev. number                */

);

/* UNDEFINE STRING */

void pundefine_string (

  Pint              ws_id,          /* workstation identifier          */
  Pint              dev             /* string dev. number              */

);

/* UNDEFINE SET */

void pundefine_set (

  Pint              ws_id,          /* workstation identifier          */
  Pint              dev             /* set dev. number                 */

);

/* DESTROY SET MEASURE */

void pdestroy_set_measure (

  Pint              ws_id,          /* workstation identifier          */
  Pint              measure_proc_id /* set measure process identifier */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* UNDEFINE COMPOSITE */

void pundefine_composite (

  Pint              ws_id,          /* workstation identifier         */
  Pint              dev             /* composite dev. number          */

);

/* DESTROY COMPOSITE MEASURE */

void pdestroy_composite_measure (

  Pint              ws_id,          /* workstation identifier         */
  Pint              measure_proc_id /* composite measure process id.  */

);

/* INITIALIZE SET 3 */

void pinit_set3 (

   Pint                  ws_id,          /* workstation identifier   */
   Pint                  dev,            /* set dev. number          */
   const Pset_measure    *init_measure,  /* initial measure          */
   Pint                  pet,            /* prompt and echo type     */
   const Plimit3         *echo_volume,   /* echo volume boundaries   */
   const Pset_data       *record         /* input data record        */

);

/* INITIALIZE SET */

void pinit_set (

   Pint                  ws_id,          /* workstation identifier   */
   Pint                  dev,            /* set dev. number          */
   const Pset_measure    *init_measure,  /* initial measure          */
   Pint                  pet,            /* prompt and echo type     */
   const Plimit          *echo_area,     /* echo area boundaries     */
   const Pset_data       *record         /* input data record        */

);

/* INITIALIZE COMPOSITE 3 */

void pinit_composite3 (

   Pint                     ws_id,          /* workstation identifier  */
   Pint                     dev,            /* composite dev. number   */
   const Pcomposite_measure *init_measure,  /* initial measure         */
   Pint                     pet,            /* prompt and echo type    */
   const Plimit3            *echo_volume,   /* echo volume boundaries  */
   const Pcomposite_data    *record         /* input data record       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INITIALIZE COMPOSITE */

void pinit_composite (

   Pint                      ws_id,          /* workstation identifier  */
   Pint                      dev,            /* composite dev. number   */
   const Pcomposite_measure *init_measure,   /* initial measure         */
   Pint                      pet,            /* prompt and echo type    */
   const Plimit              *echo_area,     /* echo area boundaries    */
   const Pcomposite_data     *record         /* input data record       */

);

/* SET SET MODE */

void pset_set_mode (

   Pint              ws_id,          /* workstation identifier      */
   Pint              dev,            /* set dev. number             */
   Pop_mode          mode,           /* operating mode              */
   Pecho_switch      echo            /* echo switch                 */

);

/* SET COMPOSITE MODE */

void pset_composite_mode (

   Pint              ws_id,          /* workstation identifier      */
   Pint              dev,            /* composite dev. number       */
   Pop_mode          mode,           /* operating mode              */
   Pecho_switch      echo            /* echo switch                 */

);

/* REQUEST SET 3 */

void preq_set3 (

   Pint                  ws_id,      /* workstation identifier      */
   Pint                  dev,        /* set dev. number             */
   Pstore                store,      /* store for measure data      */
   Pna_in_status         *status,    /* OUT non-atomic input status */
   Pset_measure          **measure   /* OUT pointer to measure data */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* REQUEST SET */

void preq_set (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */

);

/* REQUEST COMPOSITE 3 */

void preq_composite3 (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */

);

/* REQUEST COMPOSITE */

void preq_composite (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* composite dev. number       */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pcomposite_measure   **measure    /* OUT pointer to measure data */

);

/* SAMPLE SET 3 */

void psample_set3 (

   Pint                 ws_id,       /* workstation identifier      */
   Pint                 dev,         /* set dev. number             */
   Pstore               store,       /* store for measure data      */
   Pna_in_status        *status,     /* OUT non-atomic input status */
   Pset_measure         **measure    /* OUT pointer to measure data */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SAMPLE SET */

void psample_set (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    dev,         /* set dev. number             */
   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pset_measure            **measure    /* OUT pointer to measure data */

);

/*  SAMPLE COMPOSITE 3 */

void psample_composite3 (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    dev,         /* composite dev. number       */
   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pcomposite_measure      **measure    /* OUT pointer to measure data */

);

/* SAMPLE COMPOSITE */

void psample_composite (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    dev,         /* composite dev. number       */
   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pcomposite_measure      **measure    /* OUT pointer to measure data */

);

/* GET SET 3 */

void pget_set3 (

   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pset_measure            **measure    /* OUT pointer to measure data */

);

/* GET SET */

void pget_set (

   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pset_measure            **measure    /* OUT pointer to measure data */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* GET COMPOSITE 3 */

void pget_composite3 (

   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pcomposite_measure      **measure    /* OUT pointer to measure data */

);

/* GET COMPOSITE */

void pget_composite (

   Pstore                  store,       /* store for measure data      */
   Pna_in_status           *status,     /* OUT non-atomic input status */
   Pcomposite_measure      **measure    /* OUT pointer to measure data */

);

/* INITIALIZE DIRECT INTERPRETATION PICK 3 */

void pinit_di_pick3 (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    pick_type,   /* pick type                   */
   Pint                    max_sets,    /* max. pick sets to return    */
   Pecho_switch            echo_switch, /* echo switch                 */
   Pint                    pet,         /* prompt and echo type        */
   const Plimit3           *echo_vol,   /* echo volume                 */
   const Pdi_pick_data3    *pick_data,  /* pick data record            */
   Ppath_order             order        /* pick path order             */

);

/* INITIALIZE  DIRECT INTERPRETATION PICK */

void pinit_di_pick (

   Pint                    ws_id,       /* workstation identifier      */
   Pint                    pick_type,   /* pick type                   */
   Pint                    max_sets,    /* max. pick sets to return    */
   Pecho_switch            echo_switch, /* echo switch                 */
   Pint                    pet,         /* prompt and echo type        */
   const Plimit            *echo_area,  /* echo area                   */
   const Pdi_pick_data     *pick_data,  /* pick data record            */
   Ppath_order             order        /* pick path order             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* ENABLE DIRECT INTERPRETATION PICK */

void penable_di_pick (

   Pint              ws_id,          /* workstation identifier      */
   Pint              struct_id       /* structure identifier        */

);

/* DISABLE DIRECT INTERPRETATION PICK */

void pdisable_di_pick (

   Pint              ws_id           /* workstation identifier      */

);

/* TRANSFER DIRECT INTERPRETATION PICK SET */

void ptrans_di_pick_set (

   Pint              ws_id,          /* workstation identifier      */
   Pstore            store,          /* handle to Store object      */
   Pin_status        *in_status,     /* OUT pick status             */
   Ppick_path_list   **pick          /* OUT direct interpretation   */
                                     /* pick sets                   */

);

/*ATTACH LOGICAL INPUT DEVICE TO VIEW */

void pattach_lid_to_view (

   Pint           lid_ws_id,         /* LID workstation id          */
   Pin_class      dev_class,         /* LID class                   */
   Pint           in_num,            /* LID number                  */
   Pint           view_ws_id,        /* view workstation id         */
   Pint           view_ind,          /* view index                  */
   Pint           view_action_type,  /* action type                 */
   Pint           transfer_type,     /* measure transfer type       */
   Pint           view_upd_method    /* view update method          */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* DETACH LOGICAL INPUT DEVICE FROM VIEW */

void pdetach_lid_from_view (

   Pint            lid_ws_id,          /* LID workstation id          */
   Pin_class       dev_class,          /* LID class                   */
   Pint            in_num,             /* LID number                  */
   Pint            view_ws_id,         /* view workstation id         */
   Pint            view_ind,           /* view index                  */
   Pint            view_action_type,   /* view action type            */
   Pint            view_upd_method     /* view update method          */

);

/* ATTACH LOGICAL INPUT DEVICE TO LIGHT SOURCE */

void pattach_lid_to_light_src (

   Pint          lid_ws_id,            /* LID workstation id          */
   Pin_class     dev_class,            /* LID class                   */
   Pint          in_num,               /* LID number                  */
   Pint          light_ws_id,          /* light ws id                 */
   Pint          light_src_ind,        /* light source index          */
   Pint          action,               /* light action type           */
   Pint          transfer_type         /* measure transfer type       */

);

/* DETACH LOGICAL INPUT DEVICE FROM LIGHT SOURCE */

void pdetach_lid_from_light_src (

   Pint          LID_ws_id,            /* LID workstation id          */
   Pin_class     dev_class,            /* LID class                   */
   Pint          in_num,               /* LID number                  */
   Pint          light_ws_id,          /* light ws id                 */
   Pint          light_src_ind,        /* light source index          */
   Pint          action                /* light action type           */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE LINETYPE DEFINITION FACILITIES */

void pinq_linetype_def_facs (

   Pint      num_elems_appl_list,    /* # elems in appl list          */
   Pint      start_ind,              /* starting elem to return       */
   Pint      *err_ind,               /* OUT error indicator           */
   Pint      *num_definable,         /* OUT num definable linetypes   */
   Pint      *num_predefined,        /* OUT num predefined linetypes  */
   Pint      *max_dash_segments,     /* OUT max dash segs supported   */
   Pint      *offset,                /* OUT offset to definable       */
                                     /* linetypes                     */
   Pint_list *predef_def_linetypes,  /* OUT list of predefined        */
                                     /* definable linetypes           */
   Pint      *num_elems_impl_list    /* OUT # elems in impl list      */

);

/* INQUIRE PREDEFINED LINETYPE DEFINITION */

void pinq_pred_linetype_def (

   Pint         linetype,              /* linetype                    */
   Pint         num_elems_appl_list,   /* # elems in appl list        */
   Pint         starting_ind,          /* starting element to return  */
   Pint         *err_ind,              /* OUT error indicator         */
      Pfloat       *repeat_length,     /* OUT dash cycle repeat length*/
   Pfloat_list  *segment_lengths,      /* OUT list of segment lengths */
   Pint         *num_elems_impl_list   /* OUT # elems in impl list    */

);

/* INQUIRE MARKER TYPE DEFINITION FACILITIES */

void pinq_marker_type_def_facs (

   Pstore     store,                 /* handle to store object       */
   Pint       *err_ind,              /* OUT error indicator          */
   Pint       *num_definable,        /* OUT num definable marker types */
   Pint       *num_predefined,       /* OUT num predef. marker types */
   Pint       *offset,               /* OUT offset to definable marker */
                                     /* types                        */
   Pint       *max_allowable_pts,    /* OUT max allowable pts        */
   Pint_list  **shape_types,         /* OUT list of avail. shape types */
   Pint_list  **predef_marker_types  /* OUT list of predefined marker  */
                                     /* types                        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PREDEFINED MARKER TYPE DEFINITION */

void pinq_pred_marker_type_def (

   Pint           marker_type,          /* marker type                */
   Pstore         store,                /* handle to store object     */
   Pint           *err_ind,             /* OUT error indicator        */
   Pmarker_desc   **marker              /* OUT marker descriptor      */

);

/* INQUIRE SET OF WORKSTATIONS TO WHICH DIRECT INTERPRETATION STRUCTURE */
/* POSTED */

void pinq_wss_di_struct_posted (

   Pint       num_elems_appl_list, /* # of elems in appl. list       */
   Pint       starting_ind,        /* starting element to return     */
   Pint       *err_ind,            /* OUT error indicator            */
   Pint_list  *ws_id,              /* OUT list of ws identifiers     */
   Pint       *num_elems_impl_list /* OUT # of elems in impl list    */

);

/* INQUIRE LINETYPE DEFINITION */

void pinq_linetype_def (

   Pint         linetype,              /* linetype                   */
   Pint         num_elems_appl_list,   /* # elems in appl list       */
   Pint         starting_ind,          /* starting element to return */
   Pint         *err_ind,              /* OUT error indicator        */
   Pfloat       *repeat_length,        /* OUT dash cycle repeat length
                                          */
   Pfloat_list  *segment_lengths,      /* OUT list of segment lengths */
   Pint         *num_elems_impl_list   /* OUT # elems in impl list   */

);

/* INQUIRE MARKER TYPE DEFINITION */

void pinq_marker_type_def (

   Pint           marker_type,          /* marker type                */
   Pstore         store,                /* handle to store object     */
   Pint           *err_ind,             /* OUT error indicator        */
   Pmarker_desc   **marker              /* OUT marker descriptor      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE WATCH ON ELEMENT RANGE */

void pinq_watch_elem_range (

   Pint           *err_ind,          /* OUT error indicator          */
   Pwatch_enable  *flag,             /* OUT watch range enable flag */
   Pint           *str_id,           /* OUT structure identifier     */
   Pint           *elem_ptr_1,       /* OUT element pointer 1 value */
   Pint           *elem_ptr_2        /* OUT element pointer 2 value */

);

/* INQUIRE VIEW STATUS */

void pinq_view_status (

   Pint              ws_id,         /* workstation id               */
   Pint              view_ind,      /* view index                   */
   Pint              *err_ind,      /* OUT error indicator          */
   Pview_status      *view_status   /* OUT view status              */

);

/* INQUIRE VIEW REFERENCE POINT 3 */

void pinq_view_ref_point3 (

   Pint           ws_id,            /* workstation id               */
   Pint           view_ind,         /* view index                   */
   Pint           *err_ind,         /* OUT error indicator          */
   Ppoint3        *view_ref_point   /* OUT view reference point     */

);

/* INQUIRE VIEW REFERENCE POINT */

void pinq_view_ref_point (

   Pint           ws_id,            /* workstation id               */
   Pint           view_ind,         /* view index                   */
   Pint           *err_ind,         /* OUT error indicator          */
   Ppoint         *view_ref_point   /* OUT view reference point     */

);

/* INQUIRE VIEW PLANE NORMAL */

void pinq_view_plane_norm (

   Pint           ws_id,            /* workstation id               */
   Pint           view_ind,         /* view index                   */
   Pint           *err_ind,         /* OUT error indicator          */
   Pvec3          *view_norm_vec    /* OUT view plane normal        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE VIEW UP VECTOR 3 */

void pinq_view_up_vec3 (

   Pint             ws_id,              /* workstation id              */
   Pint             view_ind,           /* view index                  */
   Pint             *err_ind,           /* OUT error indicator         */
   Pvec3            *view_up_vec        /* OUT view up vector          */

);

/* INQUIRE VIEW UP VECTOR */

void pinq_view_up_vec (

   Pint             ws_id,              /* workstation id              */
   Pint             view_ind,           /* view index                  */
   Pint             *err_ind,           /* OUT error indicator         */
   Pvec             *view_up_vec        /* OUT view up vector          */

);

/* INQUIRE VIEW WINDOW LIMITS */

void pinq_view_win_limits (

   Pint               ws_id,            /* workstation id              */
   Pint               view_ind,         /* view index                  */
   Pint               *err_ind,         /* OUT error indicator         */
   Plimit             *win              /* OUT view window limits      */

);

/* INQUIRE PROJECTION VIEWPORT 3 */

void pinq_proj_vp3 (

   Pint             ws_id,            /* workstation id                 */
   Pint             view_ind,         /* view index                     */
   Pint             *err_ind,         /* OUT error indicator            */
   Plimit3          *proj_vp          /* OUT projection viewport limits */

);

/* INQUIRE PROJECTION VIEWPORT */

void pinq_proj_vp (

   Pint             ws_id,            /* workstation id                 */
   Pint             view_ind,         /* view index                     */
   Pint             *err_ind,         /* OUT error indicator            */
   Plimit           *proj_vp          /* OUT projection viewport limits */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PROJECTION TYPE */

void pinq_proj_type (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pproj_type        *proj_type      /* OUT projection type         */

);

/* INQUIRE PROJECTION REFERENCE POINT */

void pinq_proj_ref_point (

   Pint          ws_id,              /* workstation id                 */
   Pint          view_ind,           /* view index                     */
   Pint          *err_ind,           /* OUT error indicator            */
   Ppoint3       *proj_ref_point     /* OUT projection reference point */

);

/* INQUIRE VIEW PLANE DISTANCE */

void pinq_view_plane_dist (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pfloat            *view_plane     /* OUT view plane distance     */

);

/* INQUIRE FRONT PLANE DISTANCE */

void pinq_front_plane_dist (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pfloat            *front_plane    /* OUT front plane distance    */

);

/* INQUIRE BACK PLANE DISTANCE */

void pinq_back_plane_dist (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pfloat            *back_plane     /* OUT back plane distance     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE X-Y CLIPPING INDICATOR */

void pinq_xy_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *xy_clip        /* OUT x-y clipping indicator  */

);

/* INQUIRE FRONT CLIPPING INDICATOR */

void pinq_front_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *front_clip     /* OUT front clipping indicator
                                        */

);

/* INQUIRE BACK CLIPPING INDICATOR */

void pinq_back_clip_indicator (

   Pint              ws_id,          /* workstation id              */
   Pint              view_ind,       /* view index                  */
   Pint              *err_ind,       /* OUT error indicator         */
   Pclip_ind         *back_clip      /* OUT back clipping indicator */

);

/* INQUIRE LIST OF POSTING GROUPS */

void pinq_list_post_grps (

   Pint       ws_id,                /* workstation identifier      */
   Pint       num_elems_appl_list,  /* # elems in appl. list       */
   Pint       starting_ind,         /* starting elem to return     */
   Pint       *err_ind,             /* OUT error indicator         */
   Pint_list  *grp_ids,             /* OUT list of posting groups  */
   Pint       *num_elems_impl_list  /* OUT # elems in impl. list   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE POSTED STRUCTURES FROM  POSTING GROUP */

void pinq_posted_structs_from_post_grp (

   Pint                 ws_id,                 /* workstation id.       */
   Pint                 num_elems_appl_list,   /* # elems in appl list */
   Pint                 start_ind,             /* starting index        */
   Pint                 *err_ind,              /* OUT error indicator   */
   Pint_list            *struct_ids,           /* OUT list of posted    */
                                               /* structure networks    */
   Pint                 *num_elems_impl_list   /* OUT # elems in impl   */
                                               /* list                  */

);

/* INQUIRE POSTING GROUP */

void pinq_post_grp (

   Pint              ws_id,             /* workstation identifier      */
   Pint              grp_id,            /* posting group identifier    */
   Pint              *err_ind,          /* OUT error indicator         */
   Ppost_grp_status  *post_grp_status,  /* OUT posting group status    */
   Pint              *view_ind,         /* OUT default view index      */
   Pbackg_style      *backg_style,      /* OUT background style        */
   Pbackg_method     *backg_method,     /* OUT background method       */
   Pborder_indic     *border_indic,     /* OUT border indicator        */
   Pint              *border_ind        /* OUT border index            */

);

/* INQUIRE POSTED DIRECT INTERPRETATION STRUCTURE */

void pinq_posted_di_struct (

   Pint              ws_id,             /* workstation identifier      */
   Ptrav_type        trav_type,         /* traversal type              */
   Pint              *err_ind,          /* OUT error indicator         */
   Pposted_ind       *posted_ind        /* OUT posted indicator        */

);

/* INQUIRE DIRECT INTERPRETATION MODE */

void pinq_di_mode (

   Pint              ws_id,         /* workstation identifier          */
   Ptrav_type        trav_type,     /* traversal type                  */
   Pint              *err_ind,      /* OUT error indicator             */
   Pdi_mode          *mode          /* OUT direct interpretation mode */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE ASSOCIATED IMAGE RESOURCES */

void pinq_assoc_image_res (

   Pint        ws_id,                 /* workstation identifier          */
   Pint        num_elems_appl_list,   /* # elems in appl list            */
   Pint        starting_ind,          /* start index                     */
   Pint        *err_ind,              /* OUT error indicator             */
   Pint_list   **image_res_ids,       /* OUT list of image resource      */
                                      /* identifiers                     */
   Pint        *num_elems_impl_list   /* OUT # elems in impl list        */

);

/* INQUIRE IMAGE RESOURCE */

void pinq_image_res (

   Pint                ws_id,          /* workstation identifier      */
   Pint                image_res_id,   /* image resource identifier   */
   Pstore              store,          /* store                       */
   Pint                *err_ind,       /* OUT error indicator         */
   Pint_size           *dims,          /* OUT m,n dimensions          */
   Pimage_res          **image_res,    /* OUT image resource          */
   Paccess_flag        *access_flag    /* OUT access flag             */

);

/* INQUIRE PICTURE STATUS */

void pinq_pict_status (

   Pint                ws_id,          /* workstation identifier      */
   Ptrav_type          trav_type,      /* traversal type              */
   Pint                *err_ind,       /* OUT error indicator         */
   Ppict_stat          *pict_stat      /* OUT picture status          */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE TARGET STATE */

void pinq_targ_st (

   Pint                ws_id,          /* workstation identifier      */
   const Ptarg_addr    *targ_addr,     /* target address              */
   Pstore              store,          /* handle to Store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Pint                *offset,        /* OUT offset from base target */
   Ptarg_data_st       *st_nr_pict,    /* OUT state of non-retained   */
                                       /* picture                     */
   Ptarg_data_st       *st_ret_pict,   /* OUT state of retained       */
                                       /* picture                     */
   Pvisual_st          *visual_st,     /* OUT target state of         */
                                       /* visual rep.                 */
   Ptarg_empty_st      *empty_st,      /* OUT target empty            */
   Ptarg_type_list     **targ_types    /* OUT target usage list       */

);

/* INQUIRE EXTENDED PATTERN REPRESENTATION */

void pinq_ext_pat_rep (

   Pint                ws_id,          /* workstation identifier      */
   Pint                index,          /* pattern index               */
   Pinq_type           type,           /* type of returned value      */
   Pstore              store,          /* handle to store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Ppattern            **pattern       /* OUT pattern data            */

);

/* INQUIRE HIGHLIGHTING REPRESENTATION */

void pinq_highl_rep (

   Pint                ws_id,          /* workstation identifier      */
   Pint                index,          /* highlighting index          */
   Pinq_type           type,           /* type of returned value      */
   Pstore              store,          /* handle to store object      */
   Pint                *err_ind,       /* OUT error indicator         */
   Phighl_method       **method        /* OUT highlighting method data*/

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE LIST OF HIGHLIGHTING INDICES */

void pinq_list_highl_inds (

   Pint         ws_id,                  /* workstation identifier      */
   Pint         num_elems_appl_list,    /* # elems in appl list        */
   Pint         starting_ind,           /* starting element to return  */
   Pint         *err_ind,               /* OUT error indicator         */
   Pint_list    *highl_inds,            /* OUT list of highl indices   */
   Pint         *num_elems_impl_list    /* OUT # elems in impl list    */

);

/* INQUIRE TEXTURE REPRESENTATION */

void pinq_texture_rep (

   Pint              ws_id,          /* workstation id              */
   Pint              index,          /* texture index               */
   Pinq_type         type,           /* type of returned values     */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Ptexture_rep      **rep           /* OUT texture representation  */

);

/* INQUIRE LIST OF TEXTURE  INDICES */

void pinq_list_texture_inds (

   Pint         ws_id,                  /* workstation id              */
   Pint         num_elems_appl_list,    /* num elems in appl list      */
   Pint         start_ind,              /* start elem index            */
   Pint         *err_ind,               /* OUT error indicator         */
   Pint_list    *indices,               /* OUT list of defined texture */
                                        /* indices                     */
   Pint         *num_elems_impl_list    /* OUT num elems in impl list  */

);

/* INQUIRE TRANSPARENCY MODE */

void pinq_transparency_mode (

   Pint              ws_id,          /* workstation id              */
   Pint              *err_ind,       /* OUT error indicator         */
   Pint              *mode           /* OUT transparency mode       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE TRANSPARENCY THRESHOLDS */

void pinq_transparency_thresholds (

   Pint         ws_id,                  /* workstation id                */
   Pint         num_elems_appl_list,    /* # elems in appl list          */
   Pint         start_ind,              /* start elem index              */
   Pint         *err_ind,               /* OUT error indicator           */
   Pfloat_list  *thresholds,            /* OUT list of thresholds        */
   Pint         *num_elems_impl_list    /* OUT # elems in impl list      */

);

/* INQUIRE APPLICATION FILTER */

void pinq_appl_filter (

   Pint              ws_id,             /* workstation identifier        */
   Pint              filter_sel,        /* filter selection              */
   Pstore            store,             /* handle to store object        */
   Pint              *err_ind,          /* OUT error indicator           */
   Pfilter           **appl_filter      /* OUT application filter        */

);

/* INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 3 */

void pinq_di_pick_corr_point3 (

   Pint              ws_id,             /* workstation identifier        */
   Pint              *err_ind,          /* OUT error indicator           */
   Ppoint3           *point             /* OUT pick correlation point    */

);

/* INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT */

void pinq_di_pick_corr_point (

   Pint              ws_id,             /* workstation identifier        */
   Pint              *err_ind,          /* OUT error indicator           */
   Ppoint            *point             /* OUT pick correlation point    */

);

/* INQUIRE DIRECT INTERPRETATION PICK SET STATUS */

void pinq_di_pick_set_status (

   Pint              ws_id,             /* workstation identifier        */
   Pint              *err_ind,          /* OUT error indicator           */
   Ppick_stat        *pick_stat         /* OUT maximum number of pick    */
                                        /* reports reached               */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEVICE COORDINATE CLIP REGIONS 3 */

void pinq_dc_clip_regions3 (

   Pint              ws_id,          /* workstation identifier      */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Plimit3_list      **limit_list    /* OUT list of clip regions    */

);

/* INQUIRE DEVICE COORDINATE CLIP REGIONS */

void pinq_dc_clip_regions (

   Pint              ws_id,          /* workstation identifier      */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Plimit_list       **limit_list    /* OUT list of clip regions    */

);

/* INQUIRE TARGET MANIPULATION MODE */

void pinq_targ_manip_mode (

   Pint           ws_id,             /* workstation identifier      */
   Pint           *err_ind,          /* OUT error indicator         */
   Pint           *targ_manip_mode   /* OUT target manipulation mode
                                        */

);

/* INQUIRE TARGET DISPOSITION */

void pinq_targ_dispos (

   Pint              ws_id,          /* workstation identifier      */
   Pstore            store,          /* handle to Store object      */
   Pint              *err_ind,       /* OUT error indicator         */
   Ptarg_op_list     **targ_ops      /* OUT target operations list  */

);

/* INQUIRE DISPLAY TARGET */

void pinq_disp_targ (

   Pint              ws_id,          /* workstation identifier      */
   Pint              *err_ind,       /* OUT error indicator         */
   Pint              *offset         /* OUT offset from base target */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE RENDERING TARGET */

void pinq_rend_targ (

   Pint               ws_id,          /* workstation identifier      */
   Pint               *err_ind,       /* OUT error indicator         */
   Pint               *offset         /* OUT offset from base target */

);

/* INQUIRE TRAVERSAL RESOURCES ASSOCIATED WITH TARGET */

void pinq_trav_res_assoc_targ (

   Pint              ws_id,               /* workstation identifier   */
   const Ptarg_addr  *targ_addr,          /* target address           */
   Pint              num_elems_appl_list, /* # elems in appl list     */
   Pint              starting_ind,        .* start index             */
   Pint              *err_ind,            /* OUT error indicator      */
   Pint_list         *res_ids,            /* OUT list of associated   */
                                          /* traversal resources      */
   Pint              *num_elems_impl_list /* OUT # elems in impl list */

);

/* INQUIRE TARGET ASSOCIATED WITH TRAVERSAL RESOURCE */

void pinq_targ_assoc_trav_res (

   Pint             ws_id,          /* workstation identifier         */
   Pint             res_id,         /* traversal resource identifier  */
   Pint             *err_ind,       /* OUT error indicator            */
   Passoc_flag      *assoc_flag,    /* OUT association flag           */
   Pint             *offset         /* OUT offset from base target    */

);

/* INQUIRE LISTS OF AVAILABLE LOGICAL INPUT DEVICES */

void pinq_lists_avail_lids (

   Pint       ws_id,              /* workstation identifier           */
   Pstore     store,              /* handle to Store object           */
   Pint       *err,               /* OUT error indicator              */
   Pint_list  **avail_loc_dev,    /* OUT list available locator LIDs  */
   Pint_list  **avail_stroke_dev, /* OUT list available stroke LIDs   */
   Pint_list  **avail_val_dev,    /* OUT list available valuator LIDs */
   Pint_list  **avail_choice_dev, /* OUT list available choice LIDs   */
   Pint_list  **avail_pick_dev,   /* OUT list available pick LIDs     */
   Pint_list  **avail_string_dev, /* OUT list available string LIDs   */
   Pint_list  **avail_set_dev,    /* OUT list available set LIDs      */
   Pint_list  **avail_compos_dev  /* OUT list available composite LIDs*/

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE LOGICAL INPUT DEVICE DEFINITION */

void pinq_lid_def (

   Pint                  ws_id,          /* workstation identifier   */
   Pin_class             dev_class,      /* input class              */
   Pint                  device,         /* device number            */
   Pstore                store,          /* handle to Store object   */
   Pint                  *err,           /* OUT error indicator      */
   Pint                  *meas_proc,     /* OUT measure process      */
   Ptrig_process_list    **trig_procs,   /* OUT trigger processes    */
   Pecho_process_list    **echo_procs,   /* OUT echo processes       */
   Pack_process_list     **ack_procs     /* OUT ack. processes       */

);

/* INQUIRE SET MEASURE DEFINITION */

void pinq_set_meas_def (

   Pint                   ws_id,        /* workstation identifier    */
   Pint                   meas_id,      /* set measure process id    */
   Pint                   *err,         /* OUT error indicator       */
   Pint                   *max_meas,    /* OUT max measures in set   */
   Pmeas_process          *meas_proc    /* OUT base measure process  */

);

/* INQUIRE COMPOSITE MEASURE DEFINITION */

void pinq_composite_meas_def (

   Pint                   ws_id,        /* workstation identifier    */
   Pint                   meas_id,      /* composite measure proc id */
   Pstore                 store,        /* store for measure procs   */
   Pint                   *err,         /* OUT error indicator       */
   Pmeas_process_list     **meas_procs  /* OUT measure processes     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE SET DEVICE STATE 3 */

void pinq_set_st3 (

   Pint                ws_id,              /* workstation identifier     */
   Pint                dev,                /* set device number          */
   Pstore              store,              /* store for measure data     */
   Pint                *err,               /* OUT error indicator        */
   Pop_mode            *mode,              /* OUT operating mode         */
   Pecho_switch        *echo,              /* OUT echo switch            */
   Pfilter             **filter,           /* OUT set pick filter        */
   Pset_measure        **measure,          /* OUT initial measure        */
   Pint                *pet,               /* OUT prompt and echo type   */
   Plimit3             *echo_vol,          /* OUT echo volume            */
   Pset_data           **set_data          /* OUT input data record      */

);

/* INQUIRE SET DEVICE STATE */

void pinq_set_st (

   Pint                ws_id,              /* workstation identifier     */
   Pint                dev,                /* set device number          */
   Pstore              store,              /* store for measure data     */
   Pint                *err,               /* OUT error indicator        */
   Pop_mode            *mode,              /* OUT operating mode         */
   Pecho_switch        *echo,              /* OUT echo switch            */
   Pfilter             **filter,           /* OUT set pick filter        */
   Pset_measure        **measure,          /* OUT initial measure        */
   Pint                *pet,               /* OUT prompt and echo type   */
   Plimit              *echo_area,         /* OUT echo area              */
   Pset_data           **set_data          /* OUT input data record      */

);

/* INQUIRE COMPOSITE DEVICE STATE 3 */

void pinq_composite_st3 (

   Pint                ws_id,              /* workstation identifier     */
   Pint                dev,                /* composite dev. number      */
   Pstore              store,              /* store for measure data     */
   Pint                *err,               /* OUT error indicator        */
   Pop_mode            *mode,              /* OUT operating mode         */
   Pecho_switch        *echo,              /* OUT echo switch            */
   Pfilter             **filter,           /* OUT composite pick filter  */
   Pcomposite_measure  **measure,          /* OUT initial measure        */
   Pint                *pet,               /* OUT prompt and echo type   */
   Plimit3             *echo_vol,          /* OUT echo volume            */
   Pcomposite_data     **composite_data    /* OUT input data record      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE COMPOSITE DEVICE STATE */

void pinq_composite_st (

   Pint                ws_id,              /* workstation identifier     */
   Pint                dev,                /* composite dev. number      */
   Pstore              store,              /* store for measure data     */
   Pint                *err,               /* OUT error indicator        */
   Pop_mode            *mode,              /* OUT operating mode         */
   Pecho_switch        *echo,              /* OUT echo switch            */
   Pfilter             **filter,           /* OUT composite pick filter  */
   Pcomposite_measure  **measure,          /* OUT initial measure        */
   Pint                *pet,               /* OUT prompt and echo type   */
   Plimit              *echo_area,         /* OUT echo area              */
   Pcomposite_data     **composite_data    /* OUT input data record      */

);

/* INQUIRE DIRECT INTERPRETATION PICK STATE 3 */

void pinq_di_pick_st3 (

   Pint                ws_id,              /* workstation identifier     */
   Pinq_type           type,               /* type of returned value     */
   Pstore              store,              /* store for measure data     */
   Pint                *err,               /* OUT error indicator        */
   Pecho_switch        *echo_switch,       /* OUT echo switch            */
   Pfilter             **pick_filter,      /* OUT di pick filter         */
   Ppoint3             *correlation_pt,    /* OUT pick correlation point */
   Pint                *pick_type,         /* OUT pick type              */
   Pint                *max_sets,          /* OUT max sets to return     */
   Pin_status          *init_status,       /* OUT initial status         */
   Ppick_path_list     **init_pick_sets,   /* OUT initial di pick sets   */
   Pint                *pet,               /* OUT prompt and echo type   */
   Plimit3             *echo_vol,          /* OUT echo volume            */
   Ppick_data          **pick_data,        /* OUT input data record      */
   Ppath_order         *order              /* OUT pick path order        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE DIRECT INTERPRETATION PICK STATE */

void pinq_di_pick_st (

   Pint              ws_id,             /* workstation identifier      */
   Pinq_type         type,              /* type of returned value      */
   Pstore            store,             /* store for measure data      */
   Pint              *err,              /* OUT error indicator         */
   Pecho_switch      *echo_switch,      /* OUT echo switch             */
   Pfilter           **pick_filter      /* OUT di pick filter          */
   Ppoint            *correlation_pt    /* OUT pick correlation point  */
   Pint              *pick_type         /* OUT pick type               */
   Pint              *max_sets          /* OUT max sets to return      */
   Pin_status        *init_status       /* OUT initial status          */
   Ppick_path_list   **init_pick_sets   /* OUT initial di pick sets    */
   Pint              *pet,              /* OUT prompt and echo type    */
   Plimit            *echo_area,        /* OUT echo area               */
   Ppick_data        **pick_data        /* OUT input data record       */
   Ppath_order       *order             /* OUT pick path order         */

);

/* INQUIRE LOGICAL INPUT DEVICE ATTACHED TO VIEW */

void pinq_lid_attached_to_view (

   Pint         ws_id,          /* workstation identifier                */
   Pin_class    dev_class,      /* input class                           */
   Pint         dev,            /* logical input dev. number             */
   Pstore       store,          /* store for list of attachment data     */
   Pint         *err_ind,       /* OUT error indicator                   */
   Pint         *num_attached,  /* OUT num view attachments for LID      */
   Pview_attach **attached      /* OUT list of view attachments for LID*/

);

/* INQUIRE LOGICAL INPUT DEVICE ATTACHED TO LIGHT SOURCE */

void pinq_lid_attached_to_light_src (

   Pint           ws_id,          /* workstation identifier              */
   Pin_class      dev_class,      /* input class                         */
   Pint           dev,            /* logical input dev. number           */
   Pstore         store,          /* store for list of attachment data   */
   Pint           *err_ind,       /* OUT error indicator                 */
   Pint           *num_attached,  /* OUT num light source attachments    */
                                  /* for LID                             */
   Plight_attach **attached       /* OUT list of light source            */
                                  /* attachments for LID                 */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE CONDITIONAL TRAVERSAL FACILITIES */

void pinq_cond_trav_facs (

   Pint      ws_type,              /* workstation type                  */
   Pint      num_elems_appl_list,  /* # elems in appl list              */
   Pint      start_ind,            /* starting index                    */
   Pint      *err_ind,             /* OUT error indicator               */
   Pint      *invis_filters,       /* OUT # of invis. filters           */
   Pint      *highl_filters,       /* OUT number of highl. filters      */
   Pint      *num_pick_filters,    /* OUT number of pick filters        */
   Pint      *num_appl_filters,    /* OUT number of appl. filters       */
   Pint      *num_appl_int,        /* OUT number of appl. integers      */
   Pint      *num_appl_real,       /* OUT number of appl. reals         */
   Pint_list **test_methods,       /* OUT list of avail. test methods   */
   Pint      *num_elems_impl_list  /* OUT # elems in impl list          */

);

/* INQUIRE DIRECT INTERPRETATION TRAVERSAL FACILITIES */

void pinq_di_trav_facs (

   Pint     ws_type,                 /* workstation type                */
   Pint     *err_ind,                /* OUT error indicator             */
   Pint     *num_post_procs,         /* OUT number of simultaneously    */
                                     /* posted traversal processes      */
   Pint     *num_rend_multi_pass,    /* OUT number of rendering multi-  */
                                     /* pass algorithms supported       */
   Pint     *num_pick_multi_pass     /* OUT number of picking multi-    */
                                     /* pass algorithms supported       */

);

/* INQUIRE DEVICE COORDINATE CLIP REGIONS FACILITIES */

void pinq_dc_clip_regions_facs (

   Pint       ws_type,                 /* workstation type              */
   Pint       *err_ind,                /* OUT error indicator           */
   Pint       *num_dc_clip_regions     /* OUT number of simultaneously
                                          */
                                       /* supported DC clip regions     */

);

/* INQUIRE TARGET FACILITIES */

void pinq_targ_facs (

   Pint          ws_type,          /* workstation type                  */
   Pint          *err_ind,         /* OUT error indicator               */
   Pint          *num_targs        /* OUT number of available targets   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEFAULT TARGET DISPOSITION */

void pinq_def_targ_disp (

   Pint          ws_type,          /* workstation type                */
   Pstore        store,            /* handle to Store object          */
   Pint          *err_ind,         /* OUT error indicator             */
   Ptarg_op_list **targ_ops        /* OUT list of target operations   */

);

/* INQUIRE PREDEFINED ASSOCIATION OF TARGET WITH TRAVERSAL RESOURCES */

void pinq_pred_assoc_targ_trav_res (

   Pint              ws_type,             /* workstation type          */
   const Ptarg_addr  *targ_addr,          /* target address            */
   Pint              num_elems_appl_list, /* # elems in appl. list     */
   Pint              start_ind,           /* start element to return   */
   Pint              *err_ind,            /* OUT error indicator       */
   Pint_list         **res_ids,           /* OUT list of associated    */
                                          /* traversal resource ids    */
   Pint              *num_elems_impl_list /* OUT # elems in impl list  */

);

/* INQUIRE TRAVERSAL RESOURCE FACILITIES */

void pinq_trav_res_facs (

   Pint             ws_type,     /* workstation type                  */
   Pstore           store,       /* handle to Store object            */
   Pint             *err_ind,    /* OUT error indicator               */
   Ptrav_res_list   **trav_res   /* OUT list of available traversal   */
                                 /* resources                         */

);

/* INQUIRE PREDEFINED ASSOCIATION OF TRAVERSAL RESOURCE WITH TARGET */

void pinq_pred_assoc_trav_res_targ (

   Pint             ws_type,         /* workstation type              */
   Pint             res_id,          /* traversal resource identifier */
   Pint             *err_ind,        /* OUT error indicator           */
   Passoc_flag      *assoc_flag,     /* OUT association flag          */
   Pint             *offset          /* OUT offset from base target   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE POSTING GROUP FACILITIES */

void pinq_post_grp_facs (

   Pint         ws_type,          /* workstation type                  */
   Pstore       store,            /* handle to store object            */
   Pint         *err_ind,         /* OUT error indicator               */
   Pint         *num_def,         /* OUT num definable posting groups  */
   Pint         *num_predef,      /* OUT num predefined posting groups */
   Pint_list    **predef_grps,    /* OUT list of predefined posting    */
                                  /* groups                            */
   Pint_list    **backg_methods   /* OUT list of supported             */
                                  /* background methods                */

);

/* INQUIRE PREDEFINED POSTING GROUP */

void pinq_pred_post_grp (

   Pint              ws_type,          /* workstation type            */
   Pint              grp_id,           /* posting group identifier    */
   Pint              *err_ind,         /* OUT error indicator         */
   Ppost_grp_status  *post_grp_status, /* OUT posting group status    */
   Pint              *view_ind,        /* OUT default view index      */
   Pbackg_style      *backg_style,     /* OUT background style        */
   Pbackg_method     *backg_method,    /* OUT background method       */
   Pborder_indic     *border_indic,    /* OUT border indicator        */
   Pint              *border_ind       /* OUT border index            */

);

/* INQUIRE IMAGE RESOURCE FACILITIES */

void pinq_image_res_facs (

   Pint        ws_type,               /* workstation type             */
   Pstore      store,                 /* handle to store object       */
   Pint        *err_ind,              /* OUT error indicator          */
   Pint_list   **image_spec_methods,  /* OUT list of supported image  */
                                      /* specification methods        */
   Pint        *num_def,              /* OUT num definable image res  */
   Pint_list   **predef_res           /* OUT list of predefined image */
                                      /* resource identifiers         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PREDEFINED IMAGE RESOURCE */

void pinq_pred_image_res (

   Pint              ws_type,          /* workstation type              */
   Pint              image_res_id,     /* image resource identifier     */
   Pstore            store,            /* handle to store object        */
   Pint              *err_ind,         /* OUT error indicator           */
   Pint_size         *dims,            /* OUT m,n dimensions            */
   Pimage_res        **image_res,      /* OUT image resource            */
   Paccess_flag      *access_flag      /* OUT access flag               */

);

/* INQUIRE DYNAMICS OF POSTING GROUPS */

void pinq_dyns_post_grps (

   Pint               ws_type,         /* workstation type              */
   Pint               *err_ind,        /* OUT error indicator           */
   Pdyns_post_grps    *dyns            /* OUT posting dynamics          */

);"

/* INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE */

void pinq_dyns_ws_attrs_texture (

   Pint                 ws_type,       /* workstation type              */
   Pint                 *err_ind,      /* OUT error indicator           */
   Pdyn_ws_attrs_texture *dyns         /* OUT texture dynamics          */

);

/* INQUIRE LINETYPE DEFINTION SUPPORT */

void pinq_linetype_def_support (

   Pint              ws_type,          /* workstation type              */
   Pfloat            repeat_length,    /* repeat length                 */
   Pfloat_list       segment_lengths,  /* segment lengths               */
   Pint              *err_ind,         /* error indicator               */
   Psupport_indication*supported       /* support indication            */

);

/* INQUIRE MARKER TYPE DEFINITION SUPPORT */

void pinq_marker_type_def_support (

   Pint              ws_type,          /* workstation type              */
   const Pmarker_desc *marker,         /* marker descriptor             */
   Pint              *err_ind,         /* error indicator               */
   Psupport_indication*supported       /* support indication            */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE EXTENDED PATTERN FACILITIES */

void pinq_ext_pat_facs (

   Pint       ws_type,                /* workstation type              */
   Pstore     store,                  /* handle to store object        */
   Pint       *err_ind,               /* OUT error indicator           */
   Pint_list  **pattern_types,        /* OUT list of supported pattern*/
                                      /* types                         */
   Pint_list  **image_spec_methods    /* OUT list of supported pattern*/
                                      /* image specification methods   */

);

/* INQUIRE PREDEFINED EXTENDED PATTERN REPRESENTATION */

void pinq_pred_ext_pat_rep (

   Pint               ws_type,        /* workstation type              */
   Pint               index,          /* predefined pattern index      */
   Pstore             store,          /* handle to store object        */
   Pint               *err_ind,       /* OUT error indicator           */
   Ppattern           **pattern       /* OUT pattern data              */

);

/* INQUIRE HIGHLIGHTING FACILITIES */

void pinq_highl_facs (

   Pint         ws_type,              /* workstation type              */
   Pint         num_elems_appl_list,  /* # elems in appl list          */
   Pint         *err_ind,             /* OUT error indicator           */
   Phighl_facs  *fac,                 /* OUT highlighting facilities   */
   Pint         *num_elems_impl_list  /* OUT # elems in impl list      */

);

/* INQUIRE PREDEFINED HIGHLIGHTING REPRESENTATION */

void pinq_pred_highl_rep (

   Pint               ws_type,        /* workstation type              */
   Pint               index,          /* predefined highl. index       */
   Pstore             store,          /* handle to store object        */
   Pint               *err_ind,       /* OUT error indicator           */
   Phighl_method      **method        /* OUT highl method data         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE TEXTURE MAPPING FACILITIES */

void pinq_texture_map_facs (

   Pint               ws_type,            /* workstation type          */
   Pstore             store,              /* handle to Store object    */
   Pint               *err_ind,           /* OUT error indicator       */
   Ptexture_map_facs  **texture_map_facs  /* OUT texture facilities    */

);

/* INQUIRE TEXTURE  FACILITIES */

void pinq_texture_facs (

   Pint               ws_type,          /* workstation type           */
   Pstore             store,            /* handle to Store object     */
   Pint               *err_ind,         /* OUT error indicator        */
   Ptexture_facs      **texture_facs    /* OUT texture facilities     */

);

/* INQUIRE PREDEFINED TEXTURE REPRESENTATION */

void pinq_pred_texture_rep (

   Pint           ws_type,          /* workstation type               */
   Pint           index,            /* texture index                  */
   Pstore         store,            /* handle to Store object         */
   Pint           *err_ind,         /* OUT error indicator            */
   Ptexture_rep   **rep             /* OUT texture representation     */

);

/* INQUIRE MIPMAP FACILITIES */

void pinq_mipmap_facs (

   Pint           ws_type,          /* workstation type               */
   Pint           *err_ind,         /* OUT error indicator            */
   Pmipmap_facs   *facs             /* OUT mipmap facilities          */

);

/* INQUIRE REFLECTANCE PROPERTY FACILITIES */

void pinq_refl_prop_facs (

   Pint         ws_type,               /* workstation type           */
   Pint         num_elems_appl_list,   /* # elems in appl list       */
   Pint         start_pos,             /* starting position          */
   Pint         *err_ind,              /* OUT error indicator        */
   Pint_list    *properties,           /* OUT refl. prop. facilities */
   Pint         *num_elems_impl_list   /* OUT # elems in impl list   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE ALPHA FACILITIES */

void pinq_alpha_facs (

   Pint           ws_type,            /* workstation type            */
   Pstore         store,              /* handle to Store object      */
   Pint           *err_ind,           /* OUT error indicator         */
   Palpha_facs    **alpha_srcs        /* OUT alpha facilities        */

);

/* INQUIRE WORKSTATION STATE TABLE LENGTHS REGISTERED */

void pinq_ws_st_table_reg (

   Pint                   ws_type,    /* workstation type            */
   Pint                   *err_ind,   /* OUT error indicator         */
   Pws_st_tables_reg      *lengths    /* OUT lengths of workstation  */
                                      /* tables                      */

);

/* INQUIRE WORKSTATION STATE TABLE LENGTHS TEXTURE */

void pinq_ws_st_table_texture (

   Pint                   ws_type,    /* workstation type            */
   Pint                   *err_ind,   /* OUT error indicator         */
   Pws_st_tables_texture  *lengths    /* OUT lengths of workstation  */
                                      /* tables                      */

);

/* INQUIRE NUMBER OF AVAILABLE NON-ATOMIC LOGICAL INPUT DEVICES */

void pinq_num_avail_na_in (

   Pint       ws_type,           /* workstation type                */
   Pint       *err,              /* OUT error indicator             */
   Pnum_na_in *numin             /* OUT number of non-atomic LIDs   */

);

/* INQUIRE DEFAULT SET DEVICE DATA 3 */

void pinq_def_set_data3 (

   Pint           ws_type,        /* workstation type            */
   Pint           device,         /* set device number           */
   Pstore         store,          /* store for measure data      */
   Pint           *err,           /* OUT error indicator         */
   Pint_list      **pet_list,     /* OUT list prompt/echo types  */
   Plimit3        *echo_vol,      /* OUT echo volume             */
   Pset_data      **set_data      /* OUT default data record     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEFAULT SET DEVICE DATA */

void pinq_def_set_data (

   Pint              ws_type,          /* workstation type            */
   Pint              device,           /* set device number           */
   Pstore            store,            /* store for measure data      */
   Pint              *err,             /* OUT error indicator         */
   Pint_list         **pet_list,       /* OUT list prompt/echo types  */
   Plimit            *echo_area,       /* OUT echo area               */
   Pset_data         **set_data        /* OUT default data record     */

);

/* INQUIRE DEFAULT COMPOSITE DEVICE DATA 3 */

void pinq_def_composite_data3 (

   Pint              ws_type,          /* workstation type            */
   Pint              device,           /* composite device number     */
   Pstore            store,            /* store for measure data      */
   Pint              *err,             /* OUT error indicator         */
   Pint_list         **pet_list,       /* OUT list prompt/echo types  */
   Plimit3           *echo_vol,        /* OUT echo volume             */
   Pcomposite_data   **composite_data  /* OUT default data record     */

);

/* INQUIRE DEFAULT COMPOSITE DEVICE DATA */

void pinq_def_composite_data (

   Pint              ws_type,          /* workstation type            */
   Pint              device,           /* composite device number     */
   Pstore            store,            /* store for measure data      */
   Pint              *err,             /* OUT error indicator         */
   Pint_list         **pet_list,       /* OUT list prompt/echo types  */
   Plimit            *echo_area,       /* OUT echo area               */
   Pcomposite_data   **composite_data  /* OUT default data record     */

);

/* INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 3 */

void pinq_def_di_pick_data3 (

   Pint               ws_type,         /* workstation type            */
   Pstore             store,           /* store for measure data      */
   Pint               *err,            /* OUT error indicator         */
   Pint_list          **pet_list,      /* OUT list prompt/echo types  */
   Plimit3            *echo_vol,       /* OUT echo volume             */
   Pdi_pick_data      **pick_data      /* OUT default data record     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA */

void pinq_def_di_pick_data (

   Pint              ws_type,        /* workstation type            */
   Pstore            store,          /* store for measure data      */
   Pint              *err,           /* OUT error indicator         */
   Pint_list         **pet_list,     /* OUT list prompt/echo types  */
   Plimit            *echo_area,     /* OUT echo area               */
   Pdi_pick_data     **pick_data     /* OUT default data record     */

);

/* INQUIRE LOCATOR FACILITIES */

void pinq_loc_facs (

   Pint       ws_type,           /* workstation type                 */
   Pint       device,            /* locator device number            */
   Pstore     store,             /* store                            */
   Pint       *err,              /* OUT error indicator              */
   Pint       *max_simul_trig,   /* OUT max simul. trig procs        */
   Pint       *max_simul_echo,   /* OUT max simul. echo procs        */
   Pint       *max_simul_ack,    /* OUT max simul. ack. procs        */
   Pint_list  **meas_proc_ids,   /* OUT avail. locator measure procs */
   Pint_list  **trig_proc_ids,   /* OUT avail. trigger procs         */
   Pint_list  **echo_proc_ids,   /* OUT avail. echo procs            */
   Pint_list  **ack_proc_ids     /* OUT avail. ack. procs            */

);

/* INQUIRE STROKE FACILITIES */

void pinq_stroke_facs (

   Pint       ws_type,           /* workstation type                 */
   Pint       device,            /* stroke device number             */
   Pstore     store,             /* store                            */
   Pint       *err,              /* OUT error indicator              */
   Pint       *max_simul_trig,   /* OUT max simul. trig procs        */
   Pint       *max_simul_echo,   /* OUT max simul. echo procs        */
   Pint       *max_simul_ack,    /* OUT max simul. ack. procs        */
   Pint_list  **meas_proc_ids,   /* OUT avail. stroke measure procs  */
   Pint_list  **trig_proc_ids,   /* OUT avail. trigger procs         */
   Pint_list  **echo_proc_ids,   /* OUT avail. echo procs            */
   Pint_list  **ack_proc_ids     /* OUT avail. ack. procs            */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE VALUATOR FACILITIES */

void pinq_val_facs (

   Pint        ws_type,           /* workstation type                    */
   Pint        device,            /* valuator device number              */
   Pstore      store,             /* store                               */
   Pint        *err,              /* OUT error indicator                 */
   Pint        *max_simul_trig,   /* OUT max simul. trig procs           */
   Pint        *max_simul_echo,   /* OUT max simul. echo procs           */
   Pint        *max_simul_ack,    /* OUT max simul. ack. procs           */
   Pint_list   **meas_proc_ids,   /* OUT avail. valuator measure procs */
   Pint_list   **trig_proc_ids,   /* OUT avail. trigger procs            */
   Pint_list   **echo_proc_ids,   /* OUT avail. echo procs               */
   Pint_list   **ack_proc_ids     /* OUT avail. ack. procs               */

);

/* INQUIRE CHOICE FACILITIES */

void pinq_choice_facs (

   Pint        ws_type,           /* workstation type                    */
   Pint        device,            /* choice device number                */
   Pstore      store,             /* store                               */
   Pint        *err,              /* OUT error indicator                 */
   Pint        *max_simul_trig,   /* OUT max simul. trig procs           */
   Pint        *max_simul_echo,   /* OUT max simul. echo procs           */
   Pint        *max_simul_ack,    /* OUT max simul. ack. procs           */
   Pint_list   **meas_proc_ids,   /* OUT avail. choice measure procs   */
   Pint_list   **trig_proc_ids,   /* OUT avail. trigger procs            */
   Pint_list   **echo_proc_ids,   /* OUT avail. echo procs               */
   Pint_list   **ack_proc_ids     /* OUT avail. ack. procs               */

);

/* INQUIRE PICK FACILITIES */

void pinq_pick_facs (

   Pint        ws_type,           /* workstation type                    */
   Pint        device,            /* pick device number                  */
   Pstore      store,             /* store                               */
   Pint        *err,              /* OUT error indicator                 */
   Pint        *max_simul_trig,   /* OUT max simul. trig procs           */
   Pint        *max_simul_echo,   /* OUT max simul. echo procs           */
   Pint        *max_simul_ack,    /* OUT max simul. ack. procs           */
   Pint_list   **meas_proc_ids,   /* OUT avail. pick measure procs     */
   Pint_list   **trig_proc_ids,   /* OUT avail. trigger procs            */
   Pint_list   **echo_proc_ids,   /* OUT avail. echo procs               */
   Pint_list   **ack_proc_ids     /* OUT avail. ack. procs               */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE STRING FACILITIES */

void pinq_string_facs (

   Pint        ws_type,            /* workstation type                  */
   Pint        device,             /* string device number              */
   Pstore      store,              /* store                             */
   Pint        *err,               /* OUT error indicator               */
   Pint        *max_simul_trig,    /* OUT max simul. trig procs         */
   Pint        *max_simul_echo,    /* OUT max simul. echo procs         */
   Pint        *max_simul_ack,     /* OUT max simul. ack. procs         */
   Pint_list   **meas_proc_ids,    /* OUT avail. string measure procs   */
   Pint_list   **trig_proc_ids,    /* OUT avail. trigger procs          */
   Pint_list   **echo_proc_ids,    /* OUT avail. echo procs             */
   Pint_list   **ack_proc_ids      /* OUT avail. ack. procs             */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE SET FACILITIES */

void pinq_set_facs (

   Pint        ws_type,                  /* workstation type            */
   Pstore      store,                    /* store                       */
   Pint        *err,                     /* OUT error indicator         */
   Pint        *max_measures,            /* OUT max instances           */
   Pint        *max_simul_trig,          /* OUT max simul. trig. procs  */
   Pint        *max_simul_echo,          /* OUT max simul. echo procs   */
   Pint        *max_simul_ack,           /* OUT max simul. ack. procs   */
   Pint_list   **loc_meas_proc_ids,      /* OUT avail locator meas procs*/
   Pint_list   **stroke_meas_proc_ids,   /* OUT avail stroke meas procs */
   Pint_list   **val_meas_proc_ids,      /* OUT avail valuator meas     */
                                         /* procs                       */
   Pint_list   **choice_meas_proc_ids,   /* OUT avail choice meas procs */
   Pint_list   **pick_meas_proc_ids,     /* OUT avail pick meas procs   */
   Pint_list   **string_meas_proc_ids,   /* OUT avail string meas procs */
   Pint_list   **set_meas_proc_ids,      /* OUT avail set meas procs    */
   Pint_list   **comp_meas_proc_ids,     /* OUT avail composite meas    */
                                         /* procs                       */
   Pint_list   **trig_proc_ids,          /* OUT avail trigger procs     */
   Pint_list   **loc_echo_proc_ids,      /* OUT avail locator echo procs*/
   Pint_list   **stroke_echo_proc_ids,   /* OUT avail stroke echo procs */
   Pint_list   **val_echo_proc_ids,      /* OUT avail valuator echo     */
                                         /* procs                       */
   Pint_list   **choice_echo_proc_ids,   /* OUT avail choice echo procs */
   Pint_list   **pick_echo_proc_ids,     /* OUT avail pick echo procs   */
   Pint_list   **string_echo_proc_ids,   /* OUT avail string echo procs */
   Pint_list   **set_echo_proc_ids,      /* OUT avail set echo procs    */
   Pint_list   **comp_echo_proc_ids,     /* OUT avail composite echo    */
                                         /* procs                       */
   Pint_list   **ack_proc_ids            /* OUT avail acknowledgement   */
                                         /* procs                       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE COMPOSITE FACILITIES */

void pinq_composite_facs (

   Pint        ws_type,                 /* workstation type             */
   Pint        device,                  /* composite device number      */
   Pstore      store,                   /* store                        */
   Pint        *err,                    /* OUT error indicator          */
   Pint        *max_simul_meas,         /* OUT max simul. meas. procs   */
   Pint        *max_simul_trig,         /* OUT max simul. trig. procs   */
   Pint        *max_simul_echo,         /* OUT max simul. echo procs    */
   Pint        *max_simul_ack,          /* OUT max simul. ack. procs    */
   Pint_list   **loc_meas_proc_ids,     /* OUT avail locator meas procs*/
   Pint_list   **stroke_meas_proc_ids,  /* OUT avail stroke meas procs */
   Pint_list   **val_meas_proc_ids,     /* OUT avail valuator meas      */
                                        /* procs                        */
   Pint_list   **choice_meas_proc_ids,  /* OUT avail choice meas procs */
   Pint_list   **pick_meas_proc_ids,    /* OUT avail pick meas procs    */
   Pint_list   **string_meas_proc_ids,  /* OUT avail string meas procs */
   Pint_list   **set_meas_proc_ids,     /* OUT avail set meas procs     */
   Pint_list   **comp_meas_proc_ids,    /* OUT avail composite meas     */
                                        /* procs                        */
   Pint_list   **trig_proc_ids,         /* OUT avail trigger procs      */
   Pint_list   **loc_echo_proc_ids,     /* OUT avail locator echo procs*/
   Pint_list   **stroke_echo_proc_ids,  /* OUT avail stroke echo procs */
   Pint_list   **val_echo_proc_ids,     /* OUT avail valuator echo      */
                                        /* procs                        */
   Pint_list   **choice_echo_proc_ids,  /* OUT avail choice echo procs */
   Pint_list   **pick_echo_proc_ids,    /* OUT avail pick echo procs    */
   Pint_list   **string_echo_proc_ids,  /* OUT avail string echo procs */
   Pint_list   **set_echo_proc_ids,     /* OUT avail set echo procs     */
   Pint_list   **comp_echo_proc_ids,    /* OUT avail composite echo     */
                                        /* procs                        */
   Pint_list   **ack_proc_ids           /* OUT avail acknowledgement    */
                                        /* procs                        */

);

/* INQUIRE PREDEFINED SET MEASURE DEFINITION */

void pinq_pred_set_meas_def (

   Pint                 ws_type,       /* workstation type             */
   Pint                 meas_proc_id, /* set measure proc id           */
   Pint                 *err,          /* OUT error indicator          */
   Pint                 *max_measures,/* OUT max measures in set       */
   Pmeas_process        *meas_proc     /* OUT base measure process     */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PREDEFINED COMPOSITE MEASURE DEFINITION */

void pinq_pred_composite_meas_def (

   Pint                    ws_type,       /* workstation type           */
   Pint                    meas_proc_id,  /* composite measure proc id  */
   Pstore                  store,         /* store for measure data     */
   Pint                    *err,          /* OUT error indicator        */
   Pmeas_process_list      **meas_procs   /* OUT measure processes      */

);

/* INQUIRE SET OF GROUPS TO WHICH POSTED */

void pinq_grps_posted (

   Pint         ws_id,                  /* workstation identifier      */
   Pint         struct_id,              /* structure identifier        */
   Pint         num_elems_appl_list,    /* # elems in appl list        */
   Pint         start_ind,              /* starting index              */
   Pint         *err,                   /* OUT error indicator         */
   Pint_list    *grps,                  /* OUT list of posting groups  */
   Pint         *num_elems_impl_list    /* OUT # elems in impl list    */

);"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex C Macros for short function identifiers

Pages 277 to 283

C.1 Short function identifiers

Merge alphabetically by macro name the following macros for short function identifiers with those of PHIGS and PHIGS PLUS:

"

```
#define passoc_image_res                    pasimg
#define passoc_trav_res                     pastrs
#define pattach_lid_to_light_src            patls
#define pattach_lid_to_view                 patvw
#define pcircle                             pcirc
#define pcircle3                            pcirc3
#define pcircular_arc                       pcarc
#define pcircular_arc3                      pcarc3
#define pcircular_arc_close                 pcacl
#define pcircular_arc_close3                pcacl3
#define pclear_targ                         pclrtg
#define pclose_di_struct                    pcldis
#define pcond_exec_structure                pcexst
#define pcond_inst_structure                pcinst
#define pcond_return                        pcrtrn
#define pcond_skip_elems                    pcskel
#define pcond_skip_to_label                 pcsklb
#define pcopy_elem                          pcpel
#define pcopy_elem_range                    pcpelr
#define pcopy_elems_between_labels          pcpell
#define pcopy_targ                          pcptg
#define pcreate_composite_measure           pcrcms
#define pcreate_mipmap_texture              pcmmtx
#define pcreate_set_measure                 pcrsms
#define pcreate_targ                        pcrtg
#define pdefine_choice                      pdfcho
#define pdefine_composite                   pdfcmp
#define pdefine_linetype                    pdflty
#define pdefine_locator                     pdfloc
#define pdefine_marker_type                 pdfmty
#define pdefine_pick                        pdfpik
#define pdefine_post_grp                    pdfpgp
#define pdefine_set                         pdfset
#define pdefine_string                      pdfstg
#define pdefine_stroke                      pdfstr
#define pdefine_valuator                    pdfval
#define pdestroy_composite_measure          pdscms
#define pdestroy_set_measure                pdssms
#define pdestroy_targ                       pdestg
#define pdetach_lid_from_light_src          pdetls
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pdetach_lid_from_view                pdetvw
#define pdisable_di_pick                     pdidip
#define pdisassoc_image_res                  pdsimg
#define pdisassoc_trav_res                   pdstrs
#define pellipse                             pellp
#define pellipse3                            pellp3
#define pelliptical_arc                      pearc
#define pelliptical_arc3                     pearc3
#define pelliptical_arc_close                peacl
#define pelliptical_arc_close3               peacl3
#define penable_di_pick                      pendip
#define pend_watch_on_elem_range             pewelr
#define pfill_circle                         pfcir
#define pfill_circle3                        pfcir3
#define pfill_ellipse                        pfell
#define pfill_ellipse3                       pfell3
#define pget_composite                       pgtcm
#define pget_composite3                      pgtcm3
#define pget_set                             pgtse
#define pget_set3                            pgtse3
#define pinit_composite                      pincm
#define pinit_composite3                     pincm3
#define pinit_di_pick                        pidip
#define pinit_di_pick3                       pidip3
#define pinit_set                            pinse
#define pinit_set3                           pinse3
#define pinq_alpha_facs                      pqalfc
#define pinq_appl_filter                     pqapfl
#define pinq_assoc_image_res                 pqasim
#define pinq_back_clip_indicator             pqbkci
#define pinq_back_plane_dist                 pqbkds
#define pinq_choice_facs                     pqchfc
#define pinq_composite_facs                  pqcmfc
#define pinq_composite_meas_def              pqcmsd
#define pinq_composite_st                    pqcms
#define pinq_composite_st3                   pqcms3
#define pinq_cond_trav_facs                  pqctfc
#define pinq_dc_clip_regions                 pqdcc
#define pinq_dc_clip_regions3                pqdcc3
#define pinq_dc_clip_regions_facs            pqdccf
#define pinq_def_composite_dev_data          pqdcm
#define pinq_def_composite_dev_data3         pqdcm3
#define pinq_def_di_pick_data                pqddp
#define pinq_def_di_pick_data3               pqddp3
#define pinq_def_set_dev_data                pqdse
#define pinq_def_set_dev_data3               pqdse3
#define pinq_def_targ_disp                   pqdftd
#define pinq_di_mode                         pqdimd
#define pinq_di_pick_corr_point              pqdip
#define pinq_di_pick_corr_point3             pqdip3
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pinq_di_pick_set_stat                pqdips
#define pinq_di_pick_state                   pqdps
#define pinq_di_pick_state3                  pqdps3
#define pinq_disp_targ                       pqdstg
#define pinq_di_trav_facs                    pqdifc
#define pinq_dyns_post_grps                  pqdwsg
#define pinq_dyn_ws_attrs_texture            pqdwst
#define pinq_ext_pat_facs                    pqepf
#define pinq_ext_pat_rep                     pqepa
#define pinq_front_clip_indicator            pqfrci
#define pinq_front_plane_dist                pqfrds
#define pinq_grps_posted                     pqgpp
#define pinq_highl_facs                      pqhlfc
#define pinq_highl_rep                       pqhlr
#define pinq_image_res                       pqimrs
#define pinq_image_res_facs                  pqirfc
#define pinq_lid_def                         pqlidd
#define pinq_linetype_def                    pqltdf
#define pinq_linetype_def_facs               pqldfc
#define pinq_linetype_def_support            pqldfs
#define pinq_list_highl_ind                  pqlhi
#define pinq_list_post_grps                  pqlpgp
#define pinq_list_texture_nd                 pqlti
#define pinq_lists_avail_lids                pqllid
#define pinq_loc_facs                        pqlcfc
#define pinq_marker_type_def                 pqmtdf
#define pinq_marker_type_def_facs            pqmdfc
#define pinq_marker_type_def_support         pqmdfs
#define pinq_mipmap_facs                     pqmmfc
#define pinq_num_avail_na_in                 pqnain
#define pinq_pick_corr_point                 pqpcp
#define pinq_pick_corr_point3                pqpcp3
#define pinq_pick_facs                       pqpkfc
#define pinq_pict_stat                       pqpics
#define pinq_posted_di_struct                pqpdis
#define pinq_posted_structs_from_post_grp    pqpsgp
#define pinq_post_grp                        pqpgp
#define pinq_post_grp_facs                   pqpgfc
#define pinq_pred_assoc_trav_res_targ        pqptgr
#define pinq_pred_assoc_targ_trav_res        pqprtg
#define pinq_pred_composite_meas_def         pqpcmd
#define pinq_pred_ext_pat_rep                pqpepa
#define pinq_pred_highl_rep                  pqphlr
#define pinq_pred_image_res                  pqpimg
#define pinq_pred_linetype_def               pqpldf
#define pinq_pred_marker_type_def            pqpmdf
#define pinq_pred_post_grp                   pqppgp
#define pinq_pred_set_meas_def               pqpsmd
#define pinq_pred_texture_rep                pqptr
#define pinq_proj_ref_point                  pqpjrp
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pinq_proj_type                      pqpjty
#define pinq_proj_vp                        pqpjvp
#define pinq_proj_vp3                       pqpjv3
#define pinq_rend_targ                      pqrtg
#define pinq_set_facs                       pqsefc
#define pinq_set_meas_def                   pqsemd
#define pinq_set_state                      pqses
#define pinq_string_facs                    pqses3
#define pinq_stroke_facs                    pqskfc
#define pinq_targ_assoc_trav_res            pqtgtr
#define pinq_targ_dispos                    pqtgds
#define pinq_targ_facs                      pqtgfc
#define pinq_targ_manip_mode                pqtgmn
#define pinq_targ_st                        pqtgst
#define pinq_texture_facs                   pqtfc
#define pinq_texture_map_facs               pqtmfc
#define pinq_texture_rep                    pqtrp
#define pinq_transparency_mode              pqtrmd
#define pinq_transparency_thresholds        pqtrth
#define pinq_trav_res_assoc_targ            pqtrtg
#define pinq_trav_res_facs                  pqtrfc
#define pinq_val_facs                       pqvlfc
#define pinq_view_plane_dist                pqvwds
#define pinq_view_plane_norm                pqvpnm
#define pinq_view_ref_point                 pqvrp
#define pinq_view_ref_point3                pqvrp3
#define pinq_view_status                    pqvwst
#define pinq_view_up_vec                    pqvup
#define pinq_view_up_vec3                   pqvup3
#define pinq_view_win_limits                pqvwwl
#define pinq_watch_elem_range               pqwer
#define pinq_wss_di_struct_posted           pqwsdi
#define pinq_ws_st_table_length_texture     pqwtlt
#define pinst_structure                     pinst
#define pmanip_trav_res                     pmntr
#define pmap_dc_to_wc                       pmdcwc
#define pmap_dc_to_wsc                      pmdcws
#define pmap_wsc_to_dc                      pmwsdc
#define pmark_multi_pass_compl              pmkmpc
#define pmark_multi_pass_start              pmkmps
#define pmark_pass_compl                    pmkpsc
#define pmark_pass_start                    pmkpss
#define pmove_elem                          pmvel
#define pmove_elem_range                    pmvelr
#define pmove_elems_between_labels          pmvell
#define popen_di_struct                     popdis
#define ppop_st                             ppopst
#define ppost_di_struct                     ppsdis
#define ppost_struct_to_grp                 ppssgp
#define ppush_st                            ppshst
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define predraw_all_structs_from_grp        prasgp
#define predraw_all_structs_on_targ         prastg
#define prenew_di_state                     prdist
#define preq_composite                      prqcm
#define preq_composite3                     prqcm3
#define preq_set                            prqse
#define preq_set3                           prqse3
#define preset_all_trav_res                 prsatr
#define pret_num_passes_req                 prtnpr
#define pret_window_system_color            prtwsc
#define psample_composite                   psmcm
#define psample_composite3                  psmcm3
#define psample_set                         psmse
#define psample_set3                        psmse3
#define pset_active_textures                psatx
#define pset_alpha_data_sel_ind             psadsi
#define pset_alpha_src_sel                  psass
#define pset_appl_int                       psaiat
#define pset_appl_filter                    psafl
#define pset_appl_real                      psarat
#define pset_back_active_textures           psbatx
#define pset_back_clip_indicator            psbkci
#define pset_back_plane_dist                psbkds
#define pset_back_transparency              psbtrn
#define pset_composite_mode                 pscomm
#define pset_composite_pick_filter          pscpfl
#define pset_cond_flags                     pscfs
#define pset_cond_flags_from_tests          pscfst
#define pset_di_mode                        psdimd
#define pset_dc_clip_regions                psdcc
#define pset_dc_clip_regions3               psdcc3
#define pset_di_pick_corr_point             psdip
#define pset_di_pick_corr_point3            psdip3
#define pset_di_pick_filter                 psdipf
#define pset_disp_targ                      psdstg
#define pset_ext_pat_rep                    psepa
#define pset_front_clip_indicator           psfrci
#define pset_front_plane_dist               psfrpd
#define pset_highl_ind                      pshli
#define pset_highl_method                   pshlm
#define pset_highl_rep                      pshlr
#define pset_linetype_adaptability          psltad
#define pset_linetype_continuity            psltcn
#define pset_linetype_offset                psltof
#define pset_post_grp_backg_method          psgpbm
#define pset_post_grp_backg_style           psgpbs
#define pset_post_grp_border_ind            psgpbx
#define pset_post_grp_border_indicator      psgpbi
#define pset_post_grp_priority              psgppr
#define pset_post_grp_status                psgpst
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pset_proj_ref_point                pspjrp
#define pset_proj_type                     pspjty
#define pset_proj_vp                       pspjvp
#define pset_proj_vp3                      pspjv3
#define pset_rend_targ                     psrntg
#define pset_set_mode                      pssemd
#define pset_set_pick_filter               pssepf
#define pset_st_visual_rep                 psstvr
#define pset_targ_dispos                   pstgds
#define pset_targ_manip_mode               pstgmm
#define pset_targ_st_visual_rep            pstgvr
#define pset_texture_binding               pstbin
#define pset_texture_composition           pstcom
#define pset_texture_param                 pstpar
#define pset_texture_perspect_corr         pstpc
#define pset_texture_rep                   pstrp
#define pset_texture_res_opt_heur          pstroh
#define pset_texture_sampling              pstsam
#define pset_texture_sampling_freq         pstsfq
#define pset_transparency                  pstran
#define pset_transparency_mode             pstrmd
#define pset_transparency_thresholds       pstrth
#define pset_view_plane_dist               psvwds
#define pset_view_plane_norm               psvpnm
#define pset_view_ref_point                psvrp
#define pset_view_ref_point3               psvrp3
#define pset_view_up_vec                   psvup
#define pset_view_up_vec3                  psvup3
#define pset_view_win_limits               psvwwl
#define pset_watch_on_elem_range           pswelr
#define pset_xy_clip_indicator             psxyci
#define ptrans_di_pick_set                 ptrnpk
#define pundefine_choice                   puncho
#define pundefine_composite                puncom
#define pundefine_locator                  punloc
#define pundefine_pick                     punpik
#define pundefine_post_grp                 punpgp
#define pundefine_set                      punset
#define pundefine_string                   punstg
#define pundefine_stroke                   punstr
#define pundefine_valuator                 punval
#define punpost_all_struct_from_grp        punagp
#define punpost_di_struct                  pundis
#define punpost_struct_from_grp            punsgp
#define pupd_targ                          pupdtg
#define pupd_view_rep                      pupdvw
#define pws_type_create                    pwstyc
#define pws_type_destroy                   pwstyd
#define pws_type_get                       pwstyg
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pws_type_set                    pwstys
"
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex E Function Lists

Pages 297 to 303

E.1 List of functions ordered alphabetically by function name

Merge alphabetically by function name the following list of functions and change the page numbers accordingly:

"ASSOCIATE IMAGE RESOURCE 113
ASSOCIATE TRAVERSAL RESOURCE 116
ATTACH LOGICAL INPUT DEVICE TO LIGHT SOURCE 159
ATTACH LOGICAL INPUT DEVICE TO VIEW 158
CIRCLE 118
CIRCLE 3 117
CIRCULAR ARC 118
CIRCULAR ARC 3 118
CIRCULAR ARC CLOSE 120
CIRCULAR ARC CLOSE 3 120
CLEAR TARGET 114
CLOSE DIRECT INTERPRETATION STRUCTURE 138
CONDITIONAL EXECUTE STRUCTURE 139
CONDITIONAL INSTANCE STRUCTURE 139
CONDITIONAL RETURN 139
CONDITIONAL SKIP ELEMENTS 139
CONDITIONAL SKIP TO LABEL 139
COPY ELEMENT FROM STRUCTURE 140
COPY ELEMENT RANGE FROM STRUCTURE 140
COPY ELEMENTS BETWEEN LABELS FROM STRUCTURE 140
COPY TARGET 115
CREATE COMPOSITE MEASURE 149
CREATE MIPMAP TEXTURE 132
CREATE SET MEASURE 149
CREATE TARGET 115
DEFINE CHOICE 148
DEFINE COMPOSITE 150
DEFINE LINETYPE 132
DEFINE LOCATOR 147
DEFINE MARKER TYPE 133
DEFINE PICK 148
DEFINE POSTING GROUP 111
DEFINE SET 149
DEFINE STRING 148
DEFINE STROKE 147
DEFINE VALUATOR 147
DESTROY COMPOSITE MEASURE 152
DESTROY SET MEASURE 151
DESTROY TARGET 115
DETACH LOGICAL INPUT DEVICE FROM LIGHT SOURCE 159

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

DETACH LOGICAL INPUT DEVICE FROM VIEW 158
DISABLE DIRECT INTERPRETATION PICK 160
DISASSOCIATE IMAGE RESOURCE 113
DISASSOCIATE TRAVERSAL RESOURCE 116
ELLIPSE 119
ELLIPSE 3 118
ELLIPTICAL ARC 119
ELLIPTICAL ARC 3 119
ELLIPTICAL ARC CLOSE 121
ELLIPTICAL ARC CLOSE 3 121
ENABLE DIRECT INTERPRETATION PICK 160
END WATCH ON ELEMENT RANGE 141
FILL CIRCLE 120
FILL CIRCLE 3 119
FILL ELLIPSE 121
FILL ELLIPSE 3 120
GET COMPOSITE 158
GET COMPOSITE 3 157
GET SET 157
GET SET 3 157
INITIALIZE COMPOSITE 153
INITIALIZE COMPOSITE 3 153
INITIALIZE DIRECT INTERPRETATION PICK 160
INITIALIZE DIRECT INTERPRETATION PICK 3 159
INITIALIZE SET 152
INITIALIZE SET 3 152
INQUIRE ALPHA FACILITIES 189
INQUIRE APPLICATION FILTER 179
INQUIRE ASSOCIATED IMAGE RESOURCES 164
INQUIRE BACK CLIPPING INDICATOR 177
INQUIRE BACK PLANE DISTANCE 176
INQUIRE CHOICE FACILITIES 192
INQUIRE COMPOSITE DEVICE STATE 183
INQUIRE COMPOSITE DEVICE STATE 3 182
INQUIRE COMPOSITE FACILITIES 194
INQUIRE COMPOSITE MEASURE DEFINITION 182
INQUIRE CONDITIONAL TRAVERSAL FACILITIES 186
INQUIRE DEFAULT COMPOSITE DEVICE DATA 199
INQUIRE DEFAULT COMPOSITE DEVICE DATA 3 199
INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 187
INQUIRE DEFAULT DIRECT INTERPRETATION PICK DATA 3 187
INQUIRE DEFAULT SET DEVICE DATA 199
INQUIRE DEFAULT SET DEVICE DATA 3 198
INQUIRE DEFAULT TARGET DISPOSITION 185
INQUIRE DEVICE COORDINATE CLIP REGIONS 165
INQUIRE DEVICE COORDINATE CLIP REGIONS 3 165
INQUIRE DEVICE COORDINATE CLIP REGIONS FACILITIES 184
INQUIRE DIRECT INTERPRETATION MODE 170
INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 168
INQUIRE DIRECT INTERPRETATION PICK CORRELATION POINT 3 167

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DIRECT INTERPRETATION PICK SET STATUS 168
INQUIRE DIRECT INTERPRETATION PICK STATE 179
INQUIRE DIRECT INTERPRETATION PICK STATE 3 178
The memory referenced by *facs is managed by store. 186
INQUIRE DISPLAY TARGET 166
INQUIRE DYNAMICS OF POSTING GROUPS 186
INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES TEXTURE 186
INQUIRE EXTENDED PATTERN FACILITIES 188
INQUIRE EXTENDED PATTERN REPRESENTATION 173
INQUIRE FRONT CLIPPING INDICATOR 177
INQUIRE FRONT PLANE DISTANCE 176
INQUIRE HIGHLIGHTING FACILITIES 189
INQUIRE HIGHLIGHTING REPRESENTATION 177
INQUIRE IMAGE RESOURCE 172
INQUIRE IMAGE RESOURCE FACILITIES 188
INQUIRE LINETYPE DEFINITION 163
INQUIRE LINETYPE DEFINITION FACILITIES 161
INQUIRE LINETYPE DEFINITION SUPPORT 188
INQUIRE LIST OF HIGHLIGHTING INDICES 166
INQUIRE POSTED STRUCTURES FROM POSTING GROUP 170
INQUIRE LIST OF TEXTURE INDICES 167
INQUIRE LISTS OF AVAILABLE LOGICAL INPUT DEVICES 169
INQUIRE PREDEFINED SET MEASURE DEFINITION 198
INQUIRE LOGICAL INPUT DEVICE ATTACHED TO LIGHT SOURCE 184
INQUIRE LOGICAL INPUT DEVICE ATTACHED TO VIEW 183
INQUIRE LOGICAL INPUT DEVICE DEFINITION 180
INQUIRE MARKER TYPE DEFINITION 164
INQUIRE MARKER TYPE DEFINITION FACILITIES 161
INQUIRE MARKER TYPE DEFINITION SUPPORT 195
INQUIRE MIPMAP FACILITIES 190
INQUIRE NUMBER OF AVAILABLE NON-ATOMIC LOGICAL INPUT DEVICES 191
INQUIRE PICK FACILITIES 193
INQUIRE PICTURE STATUS 170
INQUIRE POSTED DIRECT INTERPRETATION STRUCTURE 172
INQUIRE POSTED STRUCTURES FROM POSTING GROUP 170
INQUIRE POSTED STRUCTURES FROM POSTING GROUP 170
INQUIRE POSTING GROUP FACILITIES 187
INQUIRE PREDEFINED ASSOCIATION OF TARGET WITH TRAVERSAL RESOURCES 195
INQUIRE PREDEFINED ASSOCIATION OF TRAVERSAL RESOURCE WITH TARGET 195
INQUIRE PREDEFINED COMPOSITE MEASURE DEFINITION 198
INQUIRE PREDEFINED EXTENDED PATTERN REPRESENTATION 196
INQUIRE PREDEFINED HIGHLIGHTING REPRESENTATION 197
INQUIRE PREDEFINED IMAGE RESOURCE 194
INQUIRE PREDEFINED LINETYPE DEFINITION 161
INQUIRE PREDEFINED MARKER TYPE DEFINITION 162
INQUIRE PREDEFINED POSTING GROUP 194
INQUIRE PREDEFINED SET MEASURE DEFINITION 198
INQUIRE PREDEFINED TEXTURE REPRESENTATION 197
INQUIRE PROJECTION REFERENCE POINT 176
INQUIRE PROJECTION TYPE 175

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PROJECTION VIEWPORT 175
INQUIRE PROJECTION VIEWPORT 3 175
INQUIRE RENDERING TARGET 166
INQUIRE SET DEVICE STATE 181
INQUIRE SET DEVICE STATE 3 181
INQUIRE SET FACILITIES 193
INQUIRE SET MEASURE DEFINITION 180
INQUIRE SET OF GROUPS TO WHICH POSTED 200
INQUIRE SET OF WORKSTATIONS TO WHICH DIRECT INTERPRETATION STRUCTURE POSTED 162
INQUIRE STRING FACILITIES 193
INQUIRE STROKE FACILITIES 192
INQUIRE TARGET ASSOCIATED WITH TRAVERSAL RESOURCE 171
INQUIRE TARGET DISPOSITION 165
INQUIRE TARGET FACILITIES 185
INQUIRE TARGET MANIPULATION MODE 165
INQUIRE TARGET STATE 171
INQUIRE TEXTURE FACILITIES 190
INQUIRE TEXTURE MAPPING FACILITIES 189
INQUIRE TEXTURE REPRESENTATION 178
INQUIRE TRANSPARENCY MODE 167
INQUIRE TRANSPARENCY THRESHOLDS 167
INQUIRE TRAVERSAL RESOURCE FACILITIES 185
INQUIRE TRAVERSAL RESOURCES ASSOCIATED WITH TARGET 171
INQUIRE VALUATOR FACILITIES 192
INQUIRE VIEW PLANE DISTANCE 176
INQUIRE VIEW PLANE NORMAL 174
INQUIRE VIEW REFERENCE POINT 174
INQUIRE VIEW REFERENCE POINT 3 173
INQUIRE VIEW STATUS 173
INQUIRE VIEW UP VECTOR 174
INQUIRE VIEW UP VECTOR 3 174
INQUIRE VIEW WINDOW LIMITS 175
Inquiry functions for workstation state list 164
INQUIRE WORKSTATION STATE TABLE LENGTHS HIGHLIGHTING 191
INQUIRE WORKSTATION STATE TABLE LENGTHS TEXTURE 190
INQUIRE X-Y CLIPPING INDICATOR 177
INSTANCE STRUCTURE 138
MANIPULATE TRAVERSAL RESOURCE 116
MAP DEVICE COORDINATES TO WORLD COORDINATES 138
MAP DEVICE COORDINATES TO WINDOW SYSTEM COORDINATES 137
MAP WINDOW SYSTEM COORDINATES TO DEVICE COORDINATES 137
MARK MULTI-PASS COMPLETION 144
MARK MULTI-PASS START 143
MARK PASS COMPLETION 144
MARK PASS START 144
MOVE ELEMENT FROM STRUCTURE 141
MOVE ELEMENT RANGE FROM STRUCTURE 141
MOVE ELEMENTS BETWEEN LABELS FROM STRUCTURE 141
OPEN DIRECT INTERPRETATION STRUCTURE 138

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

POP STATE 140
POST DIRECT INTERPRETATION STRUCTURE 142
POST STRUCTURE TO GROUP 142
PUSH STATE 140
REDRAW ALL STRUCTURES FROM POSTING GROUP 110
REDRAW ALL STRUCTURES FROM POSTING GROUP 110
RENEW DIRECT INTERPRETATION STATE 143
REQUEST COMPOSITE 155
REQUEST COMPOSITE 3 155
REQUEST SET 154
REQUEST SET 3 154
RESET ALL TRAVERSAL RESOURCES 116
RETRIEVE NUMBER OF PASSES REQUIRED 144
RETRIEVE WINDOW SYSTEM COLOUR 117
SAMPLE COMPOSITE 156
SAMPLE COMPOSITE 3 156
SAMPLE SET 156
SAMPLE SET 3 155
SET ACTIVE TEXTURES 125
SET ALPHA DATA SELECTION INDEX 125
SET ALPHA SOURCE SELECTOR 125
SET APPLICATION FILTER 132
SET APPLICATION INTEGER 117
SET APPLICATION REAL 117
SET BACK ACTIVE TEXTURES 125
SET BACK CLIPPING INDICATOR 137
SET BACK PLANE DISTANCE 136
SET BACK TRANSPARENCY 125
SET COMPOSITE MODE 154
SET COMPOSITE PICK FILTER 145
SET CONDITION FLAGS 126
SET CONDITION FLAGS FROM TESTS 126
SET DEVICE COORDINATE CLIP REGIONS 113
SET DEVICE COORDINATE CLIP REGIONS 3 113
SET DIRECT INTERPRETATION MODE 143
SET DIRECT INTERPRETATION PICK CORRELATION POINT 146
SET DIRECT INTERPRETATION PICK CORRELATION POINT 3 145
SET DIRECT INTERPRETATION PICK FILTER 145
SET DISPLAY TARGET 114
SET INTERIOR REPRESENTATION MASK 128
SET FRONT CLIPPING INDICATOR 136
SET FRONT PLANE DISTANCE 136
SET HIGHLIGHTING INDEX 122
SET HIGHLIGHTING METHOD 124
SET HIGHLIGHTING REPRESENTATION 131
SET LINETYPE ADAPTABILITY 122
SET LINETYPE CONTINUITY 122
SET LINETYPE OFFSET 122
SET POSTING GROUP BACKGROUND METHOD 112
SET POSTING GROUP BACKGROUND STYLE 112

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET POSTING GROUP BORDER INDEX 112
SET POSTING GROUP BORDER INDICATOR 112
SET POSTING GROUP PRIORITY 111
SET POSTING GROUP STATUS 111
SET PROJECTION REFERENCE POINT 135
SET PROJECTION TYPE 135
SET PROJECTION VIEWPORT 135
SET PROJECTION VIEWPORT 3 134
SET RENDERING TARGET 114
SET SET MODE 153
SET SET PICK FILTER 145
SET STATE OF VISUAL REPRESENTATION 115
SET TARGET DISPOSITION 114
SET TARGET MANIPULATION MODE 114
SET TARGET STATE OF VISUAL REPRESENTATION 115
SET TEXTURE BINDING 130
SET TEXTURE COMPOSITION 130
SET TEXTURE PARAMETRIZATION 130
SET TEXTURE PERSPECTIVE CORRECTION 126
SET TEXTURE REPRESENTATION 129
SET TEXTURE RESOURCE OPTIMIZATION HEURISTICS 126
SET TEXTURE SAMPLING 130
SET TEXTURE SAMPLING FREQUENCY 126
SET TRANSPARENCY 124
SET TRANSPARENCY MODE 131
SET TRANSPARENCY THRESHOLDS 131
SET VIEW PLANE DISTANCE 135
SET VIEW PLANE NORMAL 133
SET VIEW REFERENCE POINT 133
SET VIEW REFERENCE POINT 3 133
SET VIEW UP VECTOR 134
SET VIEW UP VECTOR 3 134
SET VIEW WINDOW LIMITS 134
SET WATCH ON ELEMENT RANGE 141
SET X-Y CLIPPING INDICATOR 136
TRANSFER DIRECT INTERPRETATION PICK SET 160
UNDEFINE CHOICE 151
UNDEFINE COMPOSITE 152
UNDEFINE LOCATOR 150
UNDEFINE PICK 151
UNDEFINE POSTING GROUP 111
UNDEFINE SET 151
UNDEFINE STRING 151
UNDEFINE STROKE 150
UNDEFINE VALUATOR 150
UNPOST ALL STRUCTURES FROM GROUP 142
UNPOST DIRECT INTERPRETATION STRUCTURE 143
UNPOST STRUCTURES FROM GROUPS 142
REDRAW ALL STRUCTURES FROM POSTING GROUP ON TARGET 110
UPDATE VIEW REPRESENTATION 137

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

WORKSTATION TYPE CREATE 109
WORKSTATION TYPE DESTROY 109
WORKSTATION TYPE GET 109
WORKSTATION TYPE SET 109
"

Pages 303 to 309

E.2 List of functions ordered alphabetically by bound name

Merge alphabetically by bound name the following list of functions and change the page numbers accordingly:

"passoc_image_res (...) 113
passoc_trav_res (...) 116
pattach_lid_to_light_src (...) 159
pattach_lid_to_view (...) 158
pcircle (...) 118
pcircle3 (...) 117
pcircular_arc (...) 118
pcircular_arc3 (...) 118
pcircular_arc_close (...) 120
pcircular_arc_close3 (...) 120
pclear_targ (...) 114
pclose_di_struct (...) 138
pcond_exec_struct (...) 139
pcond_inst_struct (...) 139
pcond_return (...) 139
pcond_skip_elements (...) 139
pcond_skip_to_label (...) 139
pcopy_elem_struct (...) 140
pcopy_elem_range_struct (...) 140
pcopy_elems_between_labels_struct (...) 140
pcopy_targ (...) 115
pcreate_composite_measure (...) 149
pcreate_mipmap_texture (...) 132
pcreate_set_measure (...) 149
pclear_targ (...) 114
pdefine_choice (...) 148
pdefine_composite (...) 150
pdefine_linetype (...) 132
pdefine_locator (...) 147
pdefine_marker_type (...) 133
pdefine_pick (...) 148
pdefine_post_grp (...) 111
pdefine_set (...) 149
pdefine_string (...) 148
pdefine_stroke (...) 147
pdefine_valuator (...) 147

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

pdestroy_composite_measure (...) 152
pdestroy_set_measure (...) 151
pclear_targ (...) 114
pdetach_lid_from_light_src (...) 159
pdetach_lid_from_view (...) 158
pdisable_di_pick (...) 160
pdisassoc_image_res (...) 113
pdisassoc_trav_res (...) 116
pellipse (...) 119
pellipse3 (...) 118
pelliptical_arc (...) 119
pelliptical_arc3 (...) 119
pelliptical_arc_close (...) 121
pelliptical_arc_close3 (...) 121
penable_di_pick (...) 160
pend_watch_on_elem_range (...) 141
pfill_circle (...) 120
pfill_circle3 (...) 119
pfill_ellipse (...) 121
pfill_ellipse3 (...) 120
pget_composite (...) 158
pget_composite3 (...) 157
pget_set (...) 157
pget_set3 (...) 157
pinit_composite (...) 153
pinit_composite3 (...) 153
pinit_di_pick (...) 160
pinit_di_pick3 (...) 159
pinit_set (...) 152
pinit_set3 (...) 152
pinq_alpha_facs (...) 189
pinq_appl_filter (...) 179
pinq_assoc_image_res (...) 164
pinq_back_clip_indicator (...) 177
pinq_back_plane_dist (...) 176
pinq_choice_facs (...) 192
pinq_composite_facs (...) 194
pinq_composite_meas_def (...) 182
pinq_composite_st (...) 183
pinq_composite_st3 (...) 182
pinq_cond_trav_facs (...) 186
pinq_dc_clip_regions (...) 165
pinq_dc_clip_regions3 (...) 165
pinq_dc_clip_regions_facs(...) 184
pinq_def_composite_data (...) 199
pinq_def_composite_data3 (...) 199
pinq_def_di_pick_data (...) 187
pinq_def_di_pick_data3 (...) 187
pinq_def_set_data (...) 199
pinq_def_set_data3 (...) 198

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

pinq_def_targ_disp (...) 185
pinq_di_mode (...) 170
pinq_di_pick_corr_point (...) 168
pinq_di_pick_corr_point3 (...) 167
pinq_di_pick_set_status (...) 168
pinq_di_pick_st (...) 179
pinq_di_pick_st3 (...) 178
pinq_disp_targ (...) 166
pinq_di_trav_facs (...) 186
pinq_dyns_post_grps (...) 186
pinq_dyns_ws_attrs_texture (...) 186
pinq_ext_pat_facs (...) 188
pinq_ext_pat_rep (...) 173
pinq_front_clip_indicator (...) 177
pinq_front_plane_dist (...) 176
pinq_grps_posted (...) 200
pinq_highl_facs (...) 189
pinq_highl_rep (...) 177
pinq_image_res (...) 172
pinq_image_res_facs (...) 188
pinq_lid_attached_to_light_src (...) 184
pinq_lid_attached_to_view (...) 183
pinq_lid_def (...) 180
pinq_linetype_def (...) 163
pinq_linetype_def_facs (...) 161
pinq_linetype_def_support (...) 188
pinq_list_highl_inds (...) 166
pinq_list_post_grps (...) 164
pinq_list_texture_inds (...) 167
pinq_lists_avail_lids (...) 169
pinq_loc_facs (...) 191
pinq_marker_type_def (...) 164
pinq_marker_type_def_facs (...) 161
pinq_marker_type_def_support (...) 195
pinq_mipmap_facs (...) 190
pinq_num_avail_na_in (...) 191
pinq_pick_facs (...) 193
pinq_pict_status (...) 170
pinq_posted_di_struct (...) 172
pinq_posted_structs_from_post_grp (...) 170
pinq_post_grp (...) 169
pinq_post_grp_facs (...) 187
pinq_pred_assoc_targ_trav_res (...) 195
pinq_pred_assoc_trav_res_targ (...) 195
pinq_pred_composite_meas_def (...) 198
pinq_pred_ext_pat_rep (...) 196
pinq_pred_highl_rep (...) 197
pinq_pred_image_res (...) 194
pinq_pred_linetype_def (...) 161
pinq_pred_marker_type_def (...) 162

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

pinq_pred_post_grp (...) 194
pinq_pred_set_meas_def (...) 198
pinq_pred_texture_rep (...) 197
pinq_proj_ref_point (...) 176
pinq_proj_type (...) 175
pinq_proj_vp (...) 175
pinq_proj_vp3 (...) 175
pinq_rend_targ (...) 166
pinq_set_facs (...) 193
pinq_set_meas_def (...) 180
pinq_set_st (...) 181
pinq_set_st3 (...) 181
pinq_string_facs (...) 193
pinq_stroke_facs (...) 192
pinq_targ_assoc_trav_res (...) 171
pinq_targ_dispos (...) 165
pinq_targ_facs (...) 185
pinq_targ_manip_mode (...) 165
pinq_targ_st (...) 171
pinq_texture_facs (...) 190
pinq_texture_map_facs (...) 189
pinq_texture_rep (...) 178
pinq_transparency_mode (...) 167
pinq_transparency_thresholds (...) 167
pinq_trav_res_assoc_targ (...) 171
pinq_trav_res_facs (...) 185
pinq_val_facs (...) 192
pinq_view_plane_dist (...) 176
pinq_view_plane_norm (...) 174
pinq_view_ref_point (...) 174
pinq_view_ref_point3 (...) 173
pinq_view_status (...) 173
pinq_view_up_vec (...) 174
pinq_view_up_vec3 (...) 174
pinq_view_win_limits (...) 175
pinq_watch_elem_range (...) 162
pinq_wss_di_struct_posted (...) 162
pinq_ws_st_table_highl (...) 191
pinq_ws_st_table_texture (...) 190
pinq_xy_clip_indicator (...) 177
pinst_struct (...) 138
pmanip_trav_res (...) 116
pmap_dc_to_wc (...) 138
pmap_dc_to_wsc (...) 137
pmap_wsc_to_dc (...) 137
pmark_multi_pass_compl (...) 144
pmark_multi_pass_start (...) 143
pmark_pass_compl (...) 144
pmark_pass_start (...) 144
pmove_elem_struct (...) 141

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

pmove_elem_range_struct (...) 141
pmove_elems_between_labels_struct (...) 141
popen_di_struct (...) 138
ppop_st (...) 140
ppost_di_struct (...) 142
ppost_struct_to_grp (...) 142
ppush_st (...) 140
predraw_all_structs_from_grp (...) 110
predraw_all_structs_on_targ (...) 110
prenew_di_state (...) 143
preq_composite (...) 155
preq_composite3 (...) 155
preq_set (...) 154
preq_set3 (...) 154
preset_all_trav_res (...) 116
pret_num_passes_req (...) 144
pret_window_system_colr (...) 117
psample_composite (...) 156
psample_composite3 (...) 156
psample_set (...) 156
psample_set3 (...) 155
pset_active_textures (...) 125
pset_alpha_data_sel_ind (...) 125
pset_alpha_src_sel (...) 125
pset_appl_filter (...) 132
pset_appl_int (...) 117
pset_appl_real (...) 117
pset_back_active_textures (...) 125
pset_back_clip_indicator (...) 137
pset_back_plane_dist (...) 136
pset_back_transparency (...) 125
pset_composite_mode (...) 154
pset_composite_pick_filter (...) 145
pset_cond_flags (...) 126
pset_cond_flags_from_tests (...) 126
pset_dc_clip_regions (...) 113
pset_dc_clip_regions3 (...) 113
pset_di_mode (...) 143
pset_di_pick_corr_point (...) 146
pset_di_pick_corr_point3 (...) 145
pset_di_pick_filter (...) 145
pset_disp_targ (...) 114
pset_ext_pat_rep (...) 128
pset_front_clip_indicator (...) 136
pset_front_plane_dist (...) 136
pset_highl_ind (...) 122
pset_highl_method (...) 124
pset_highl_rep (...) 131
pset_linetype_adapt (...) 122
pset_linetype_cont (...) 122

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

pset_linetype_offset (...) 122
pset_post_grp_backg_method (...) 112
pset_post_grp_backg_style (...) 112
pset_post_grp_border_ind (...) 112
pset_post_grp_border_indicator (...) 112
pset_post_grp_priority (...) 111
pset_post_grp_status (...) 111
pset_proj_ref_point (...) 135
pset_proj_type (...) 135
pset_proj_vp (...) 135
pset_proj_vp3 (...) 134
pset_rend_targ (...) 114
pset_set_mode (...) 153
pset_set_pick_filter (...) 145
pset_st_visual_rep (...) 115
pset_targ_dispos (...) 114
pset_targ_manip_mode (...) 114
pset_targ_st_visual_rep (...) 115
pset_texture_binding (...) 130
pset_texture_composition (...) 130
pset_texture_param (...) 130
pset_texture_perspect_corr (...) 126
pset_texture_rep (...) 129
pset_texture_res_opt_heur (...) 126
pset_texture_sampling (...) 130
pset_texture_sampling_freq (...) 126
pset_transparency (...) 124
pset_transparency_mode (...) 131
pset_transparency_thresholds (...) 131
pset_view_plane_dist (...) 135
pset_view_plane_norm (...) 133
pset_view_ref_point (...) 133
pset_view_ref_point3 (...) 133
pset_view_up_vec (...) 134
pset_view_up_vec3 (...) 134
pset_view_win_limits (...) 134
pset_watch_on_elem_range (...) 141
pset_xy_clip_indicator (...) 136
ptrans_di_pick_set (...) 160
pundefine_choice (...) 151
pundefine_composite (...) 152
pundefine_locator (...) 150
pundefine_pick (...) 151
pundefine_post_grp (...) 111
pundefine_set (...) 151
pundefine_string (...) 151
pundefine_stroke (...) 150
pundefine_valuator (...) 150
punpost_all_structs_from_grp (...) 142
punpost_di_struct (...) 143

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

punpost_structs_from_grps (...) 142
pupd_targ (...) 110
pupd_view_rep (...) 137
pws_type_create (...) 109
pws_type_destroy (...) 109
pws_type_get (...) 109
pws_type_set (...) 109
"

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

© ISO/IEC

ICS 35.060; 35.140

Price based on 355 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

9593 PT*4 91 ■ 4851903 0571644 9T0 ■

INTERNATIONAL STANDARD ISO/IEC 9593-4:1991
TECHNICAL CORRIGENDUM 1

Published 1994-08-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ELECTROTECHNIQUE INTERNATIONALE

Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 4:
C

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Infographie — Interfaces langage entre un programme d'application et son support graphique —

Partie 4: C

RECTIFICATIF TECHNIQUE 1

Technical corrigendum 1 to International Standard ISO/IEC 9593-4:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics and image processing.*

Page 30

Subclause 5.3

Pcolr_rep data type, change CIE L*u*v* to be CIELUV.

UDC 681.3.01:681.3.04(084):800.92 **Ref. No. ISO/IEC 9593-4:1991/Cor.1:1994(E)**

Descriptors : data processing, information interchange, graphic data processing, programming (computers), programming languages, C (programming language).

© **ISO/IEC 1994**

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Page 40

Subclause 5.4

Pcieluv data type, replace the existing data type with the following:

```
Pcieluv CIELUV

   typedef struct {

      Pfloat      u_prime;    /* u_prime coefficient  */
      Pfloat      v_prime;    /* v_prime coefficient  */
      Pfloat      y;          /* y (luminance)        */
   } Pcieluv;
```

Page 68

Subclause 5.4

Add the following after the TEXT PRECISION data type:

```
      Ptrunc_meth TRUNCATION_METHOD
         typedef enum {
            PTRUNC_HEAD,        /* returns head of path */
            PTRUNC_TAIL         /* returns tail of path */
         } Ptrunc_meth;
```

Page 110

Subclause 7.9

RETRIEVE PATHS TO ANCESTORS, change the third parameter from

```
         Ppath_order      order,            /*  path order          */
```

to

```
         Ptrunc_meth      trunc_method,     /*  truncation method   */
```

RETRIEVE PATHS TO DESCENDANTS, change the third parameter from

```
         Ppath_order      order,            /*  path order          */
```

to

```
         Ptrunc_meth      trunc_method,     /*  truncation method   */
```

Page 151

Subclause 7.12.7

INQUIRE PATHS TO ANCESTORS, change the second parameter from

```
         Ppath_order      order,            /*  path order          */
```

to

```
         Ptrunc_meth      trunc_method,     /*  truncation method   */
```

INQUIRE PATHS TO DESCENDANTS, change the second parameter from

```
         Ppath_order      order,            /*  path order          */
```

to

```
         Ptrunc_meth      trunc_method,     /*  truncation method   */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Page 168

Clause A.2

Add the following after the TEXT PRECISION data type:

```
typedef enum {
   PTRUNC_HEAD,          /*  returns head of path  */
   PTRUNC_TAIL           /*  returns tail of path  */
} Ptrunc_meth;
```

Page 175

Clause A.2

Pcieluv data type, replace the existing data type with the following:

```
typedef struct  {
   Pfloat       u_prime;      /*  u_prime coefficient  */
   Pfloat       v_prime;      /*  v_prime coefficient  */
   Pfloat       y;            /*  y(luminance)         */
```

Page 213

Clause A.3

RETRIEVE PATHS TO ANCESTORS, change the third parameter from

```
Ppath_order        order,              /*  path order         */
```

to

```
Ptrunc_meth        trunc_method,       /*  truncation method  */
```

Page 214

Clause A.3

RETRIEVE PATHS TO DESCENDANTS, change the third parameter from

```
Ppath_order        order,              /*  path order         */
```

to

```
Ptrunc_meth        trunc_method,       /*  truncation method  */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Page 245

Clause A.3

INQUIRE PATHS TO ANCESTORS, change the second parameter from

```
        Ppath_order         order,              /*  path order         */
```

to

```
        Ptrunc_meth         trunc_method,       /*  truncation method  */
```

INQUIRE PATHS TO DESCENDANTS, change the second parameter from

```
        Ppath_order       order,                /*  path order         */
```

to

```
        Ptrunc_meth       trunc_method,         /*  truncation method  */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564099 396 ■

INTERNATIONAL STANDARD

ISO/IEC 9593-4

First edition
1991-12-15

AMENDMENT 1
1994-05-01

Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 4:
C

AMENDMENT 1

Technologies de l'information — Infographie — Interfaces langage entre un programme d'application et son support graphique —

Partie 4: C

AMENDEMENT 1

Reference number
ISO/IEC 9593-4:1991/Amd.1:1994(E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Contents

1 Scope 1
2 Normative references 2
3 The C language binding of PHIGS 3
3.1 Conformance 3
3.2 Functions versus macros 3
3.3 Character strings 3
3.4 Function identifiers 3
3.5 Registration 3
3.6 Identifiers for graphical items 4
3.7 Return values 4
3.8 Header files 4
3.9 Memory management 4
3.9.1 Inquiry functions which return simple lists 4
3.9.2 Inquiry functions which return complex data structures 4
3.9.3 Meaning of the size of an element 4
3.10 Inquiries returning structure elements 4
3.11 Error handling 5
3.11.1 Application defined error handlers 5
3.11.2 Error codes 5
3.11.3 C specific PHIGS errors 5
3.12 Storage of two-dimensional data 5
3.12.1 Storage of matrices 5
3.12.2 Storage of colour arrays 6
3.13 Data type descriptions 6
4 Tables 7
4.1 Abbreviation policy for construction of identifiers 7
4.2 Table of abbreviations 7
4.3 Function names 8
4.3.1 List ordered alphabetically by bound name 8
4.3.2 List ordered alphabetically by PHIGS function name 10
5 Type definitions 14
6 Macro definitions 15
7 C PHIGS functions 16
8 C PHIGS PLUS type definitions 17
8.1 Mapping of PHIGS PLUS data types 17
8.2 Modifications to PHIGS data types 18
8.3 Implementation dependent PHIGS PLUS type definitions 23
8.4 Implementation independent PHIGS PLUS type definitions 33

© ISO/IEC 1994
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564101 874 ■

9 C PHIGS PLUS macro definitions 55
9.1 Function identifiers 55
9.2 Error codes 56
9.3 Miscellaneous 59
9.3.1 Colour mapping methods 59
9.3.2 Colour types 59
9.3.3 Curve approximation criteria types 60
9.3.4 Data mapping methods 60
9.3.5 Interior shading method 60
9.3.6 Light source types 60
9.3.7 Parametric surface characteristic types 60
9.3.8 Polyline shading method 61
9.3.9 Reflectance models 61
9.3.10 Reflectance properties 61
9.3.11 Rendering Colour Model 61
9.3.12 Surface approximation criteria types 61
9.3.13 Element enumeration 61
10 C PHIGS PLUS functions 62
10.1 Output primitive functions 62
10.2 Attribute specification functions 68
10.2.1 Bundled attribute selection 68
10.2.2 Individual attribute selection 69
10.2.3 Workstation attribute table definition 75
10.3 Inquiry functions 78
10.3.1 Inquiry functions for the workstation state list 78
10.3.2 Inquiry functions for the workstation description table 85
A Data types in compilation order and external functions 95
A.1 Macro definitions 95
A.2 Types in compilation order 101
A.3 External functions 125
B Example programs 144
B.1 star 144
B.2 iron 144
B.3 dyna_star 144
B.4 show_line 144
B.5 xform_pline 144
B.6 surface_trimmed 144
C Macros for short function identifiers 149
C.1 Short function identifiers 149
D Memory management 152
D.1 Introduction 152
D.2 Functions that return simple lists 152
D.3 Functions that return complex data structures 152
E Function lists 153
E.1 List of functions ordered alphabetically by function name 153
E.2 List of functions ordered alphabetically by bound name 155

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564102 700 ■

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 1 to International Standard ISO/IEC 9593-4:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564103 647 ■

Introduction

Replace the current text with the following:

Part 1 of the Programmer's Hierarchical Interactive Graphics System (PHIGS) functional description is registered as ISO/IEC 9592-1:1989. Part 1 is extended by part 4, ISO/IEC 9592-4:1991 to incorporate the effects of lighting, shading and other properties important for the display of surfaces and multidimensional data. The purpose of this part of ISO/IEC 9593 is to define a standard binding for the C computer programming language to the functionality defined in ISO/IEC 9592-1:1989 and ISO/IEC 9592-4:1991.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564104 583 ■

© ISO/IEC

Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 4:
C

AMENDMENT 1

1 Scope

Replace the text with the following:

Part 1 of the Programmer's Hierarchical Interactive Graphics System (PHIGS), ISO/IEC 9592-1, and part 4 (PHIGS PLUS), ISO/IEC 9592-4, specify a language independent nucleus of a graphics system. For integration into a programming language, PHIGS and PHIGS PLUS are embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 9593 specifies such a language dependent layer for the C language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

2 Normative references

Add the following reference:

ISO/IEC 9592-4:1992, *Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) — Part 4: Plus Lumière und Surfaces, PHIGS PLUS.*

--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564106 356 ■

3 The C language binding of PHIGS

Change the clause heading to **The C language binding of PHIGS and PHIGS PLUS.**

3.1 Conformance

Replace the last two paragraphs with the following text:

In order to conform, an implementation of the C binding of PHIGS shall implement those functions specified in ISO/IEC 9592-1. The implementation shall make visible all of the declarations in the C binding specified in clauses 5 to 7 in this part of ISO/IEC 9593.

In order to conform, an implementation of the C binding of PHIGS PLUS shall implement those functions specified in ISO/IEC 9592-1 and also those functions specified in ISO/IEC 9592-4. The implementation shall make visible all of the declarations in the C binding specified in clauses 5 to 10 in this part of ISO/IEC 9593.

Thus, for example, the syntax of the function names shall be precisely as specified in this part of ISO/IEC 9593 and the parameters shall be of the data types stated in this part of ISO/IEC 9593.

3.2 Functions versus macros

No changes.

3.3 Character strings

No changes.

3.4 Function identifiers

No changes.

3.5 Registration

No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.6 Identifiers for graphical items

No changes.

3.7 Return values

No changes.

3.8 Header files

No changes.

3.9 Memory management

No changes.

3.9.1 Inquiry functions which return simple lists

No changes.

3.9.2 Inquiry functions which return complex data structures

No changes.

3.9.3 Meaning of the size of an element

No changes.

3.10 Inquiries returning structure elements

No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.11 Error handling

3.11.1 Application defined error handlers

No changes.

3.11.2 Error codes

No changes.

3.11.3 C specific PHIGS errors

Change the heading to **C binding specific errors** *and add the following binding specific errors:*

2207 `Ignoring function, two or more vertices do not contain the same type of data`
Is issued when an output primitive is created in which one or more of the vertices are specified with a particular type of data, such as normals, and other vertices are specified without the same type of data.

3.12 Storage of two-dimensional data

No changes.

3.12.1 Storage of matrices

No changes.

--`,,`-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

3.12.2 Storage of colour arrays

Change the heading to be **Storage of arrays** *and replace the text with the following:*

The entries of the `Ppat_rep` and `Px_array` data types (where x is one of `colrv`, `colrv_ctrl_point`, `ctrl_point3`, `data_ctrl_point`, `edge_flag`, `facet`, `float`, `vertex`, and `vertex3`) shall be stored such that the $(i,j)^{th}$ element in the array is given by the following expression:

```
i + j * DX
```

where:

`i = 0,..., DX-1,`

`j = 0,..., DY-1,`

`DX` = the `size_x` value of the array's `Pint_size` field, and

`DY` = the `size_y` value of the array's `Pint_size` field.

3.13 Data type descriptions

Add the clause with the following text:

The `Pedge_flag_array` data type defines an edge visibility flag for each edge of the quadrilateral mesh. The edge visibility flags are arranged in a *MXN* array of `Pedge_flag_pairs`. For each *(i,j)* `Pedge_flag_pair`, the first element speechifies the edge flag for the edge between the *i,j* and *i+1,j* vertices and the second element speechifies the edge flag for the edge between the *i,j* and *i,j+1* vertices. The first element of the *(M,j)* `Pedge_flag_pair`, where $1 \leq j \leq N$, and second element of the *(i, N)* `Pedge_flag_pair`, where $1 \leq i \leq M$, are not used.

The `Pedge_flag_triplet_list` data type consists of a list of `Pedge_flag_triplets`; one for each triangle in a triangle set. The first element of a `Pedge_flag_triplet` specifies the edge flag between the first and the second vertices of the triangle, the second element specifies the edge flag between the second and third vertices of the triangle. The third element specifies the edge flag between the third and first vertices of the triangle.

The `Px_set` data types (where x is one of `colrv`, `contour`, `edge_flag`, `float`, `vertex`, and `vertex3`) contain a list of `Px_lists`.

The `Px_set_list` data types (where x is one of `contour` and `edge_flag`) contains a list of `Px_sets`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

4 Tables

4.1 Abbreviation policy for construction of identifiers

No changes.

4.2 Table of abbreviations

Add alphabetically, to table 1, the following abbreviations:

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
activation	act
ambient	amb
approximation	approx
attenuation	atten
between	NULL
characteristics	chars
coefficient	coeff
concentration	conc
constant	const
compute	comp
criteria	crit
culling	cull
deactivate	deact
deactivation	deact
deviation	dev
diffuse	diff
directional	dir
distinguishing	disting
exponent	exp
geometric	geom
homogeneous	homo
isoparametric	isoparam
parametric	param
positional	pos
product	prod
properties	props
quadrilateral	quad

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
reflectance	refl
rendering	rend
shading	shad
specular	spec
subdivision	subd
triangle	tri
triangular	tri
trimming	trim
with	NULL

4.3 Function names

4.3.1 List ordered alphabetically by bound name

In table 2 change the column heading PHIGS Name *to* PHIGS and PHIGS PLUS Name *and merge alphabetically by bound name, the following function names:*

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pcell_array3_plus	CELL ARRAY 3 PLUS
pfill_area_set_data	FILL AREA SET WITH DATA
pfill_area_set3_data	FILL AREA SET 3 WITH DATA
pinq_b_spline_surf_facs	INQUIRE B-SPLINE SURFACE FACILITIES
pinq_colr_map_facs	INQUIRE COLOUR MAPPING FACILITIES
pinq_colr_map_method_facs	INQUIRE COLOUR MAPPING METHOD FACILITIES
pinq_colr_map_rep	INQUIRE COLOUR MAPPING REPRESENTATION
pinq_colr_map_st	INQUIRE COLOUR MAPPING STATE
pinq_curve_facs	INQUIRE CURVE FACILITIES
pinq_data_map_facs	INQUIRE DATA MAPPING FACILITIES
pinq_data_map_rep	INQUIRE DATA MAPPING REPRESENTATION
pinq_depth_cue_facs	INQUIRE DEPTH CUE FACILITIES
pinq_depth_cue_rep	INQUIRE DEPTH CUE REPRESENTATION
pinq_direct_colr_model_facs	INQUIRE DIRECT COLOUR MODEL FACILITIES
pinq_dyns_ws_plus	INQUIRE DYNAMICS OF WORKSTATION PLUS
pinq_edge_rep_plus	INQUIRE EDGE REPRESENTATION PLUS
pinq_int_facs_plus	INQUIRE INTERIOR FACILITIES PLUS
pinq_int_rep_plus	INQUIRE INTERIOR REPRESENTATION PLUS
pinq_light_source_facs	INQUIRE LIGHT SOURCE FACILITIES
pinq_light_source_rep	INQUIRE LIGHT SOURCE REPRESENTATION
pinq_line_facs_plus	INQUIRE POLYLINE FACILITIES PLUS
pinq_line_rep_plus	INQUIRE POLYLINE REPRESENTATION PLUS
pinq_list_colr_map_inds	INQUIRE LIST OF COLOUR MAPPING INDICES

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pinq_list_data_map_inds	INQUIRE LIST OF DATA MAPPING INDICES
pinq_list_depth_cue_inds	INQUIRE LIST OF DEPTH CUE INDICES
pinq_list_light_source_inds	INQUIRE LIST OF LIGHT SOURCE INDICES
pinq_list_param_surf_inds	INQUIRE LIST OF PARAMETRIC SURFACE INDICES
pinq_list_refl_inds	INQUIRE LIST OF REFLECTANCE INDICES
pinq_marker_rep_plus	INQUIRE POLYMARKER REPRESENTATION PLUS
pinq_param_surf_facs	INQUIRE PARAMETRIC SURFACE FACILITIES
pinq_param_surf_rep	INQUIRE PARAMETRIC SURFACE REPRESENTATION
pinq_pat_rep_plus	INQUIRE PATTERN REPRESENTATION PLUS
pinq_pred_colr_map_rep	INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION
pinq_pred_data_map_rep	INQUIRE PREDEFINED DATA MAPPING REPRESENTATION
pinq_pred_depth_cue_rep	INQUIRE PREDEFINED DEPTH CUE REPRESENTATION
pinq_pred_edge_rep_plus	INQUIRE PREDEFINED EDGE REPRESENTATION PLUS
pinq_pred_int_rep_plus	INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS
pinq_pred_light_source_rep	INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION
pinq_pred_line_rep_plus	INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS
pinq_pred_marker_rep_plus	INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS
pinq_pred_param_surf_rep	INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION
pinq_pred_pat_rep_plus	INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS
pinq_pred_refl_rep	INQUIRE PREDEFINED REFLECTANCE REPRESENTATION
pinq_pred_text_rep_plus	INQUIRE PREDEFINED TEXT REPRESENTATION PLUS
pinq_refl_facs	INQUIRE REFLECTANCE FACILITIES
pinq_refl_rep	INQUIRE REFLECTANCE REPRESENTATION
pinq_rend_colr_model_facs	INQUIRE RENDERING COLOUR MODEL FACILITIES
pinq_text_rep_plus	INQUIRE TEXT REPRESENTATION PLUS
pinq_trim_curve_facs	INQUIRE TRIMMING CURVE FACILITIES
pinq_ws_st_table_length_plus	INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS
pnon_uniform_b_spline_curve3	NON-UNIFORM B-SPLINE CURVE 3
pnon_uniform_b_spline_curve3_colr	NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR
pnon_uniform_b_spline_surf3	NON-UNIFORM B-SPLINE SURFACE 3
pnon_uniform_b_spline_surf3_data	NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA
ppolyline_set3_colr	POLYLINE SET 3 WITH COLOUR
pquad_mesh_data	QUADRILATERAL MESH WITH DATA
pquad_mesh3_data	QUADRILATERAL MESH 3 WITH DATA
pset_back_data_map_ind	SET BACK DATA MAPPING INDEX
pset_back_data_map_method	SET BACK DATA MAPPING METHOD
pset_back_int_colr	SET BACK INTERIOR COLOUR
pset_back_int_ind	SET BACK INTERIOR INDEX
pset_back_int_shad_method	SET BACK INTERIOR SHADING METHOD
pset_back_int_style	SET BACK INTERIOR STYLE
pset_back_int_style_ind	SET BACK INTERIOR STYLE INDEX
pset_back_refl_ind	SET BACK REFLECTANCE INDEX
pset_back_refl_model	SET BACK REFLECTANCE MODEL
pset_back_refl_props	SET BACK REFLECTANCE PROPERTIES
pset_colr_map_ind	SET COLOUR MAPPING INDEX
pset_colr_map_rep	SET COLOUR MAPPING REPRESENTATION
pset_curve_approx_crit	SET CURVE APPROXIMATION CRITERIA

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS and PHIGS PLUS Name
pset_data_map_ind	SET DATA MAPPING INDEX
pset_data_map_method	SET DATA MAPPING METHOD
pset_data_map_rep	SET DATA MAPPING REPRESENTATION
pset_depth_cue_ind	SET DEPTH CUE INDEX
pset_depth_cue_rep	SET DEPTH CUE REPRESENTATION
pset_edge_colr	SET EDGE COLOUR
pset_edge_rep_plus	SET EDGE REPRESENTATION PLUS
pset_facet_cull_mode	SET FACET CULLING MODE
pset_facet_disting_mode	SET FACET DISTINGUISHING MODE
pset_int_colr	SET INTERIOR COLOUR
pset_int_rep_plus	SET INTERIOR REPRESENTATION PLUS
pset_int_shad_method	SET INTERIOR SHADING METHOD
pset_light_source_rep	SET LIGHT SOURCE REPRESENTATION
pset_light_source_st	SET LIGHT SOURCE STATE
pset_line_colr	SET POLYLINE COLOUR
pset_line_rep_plus	SET POLYLINE REPRESENTATION PLUS
pset_line_shad_method	SET POLYLINE SHADING METHOD
pset_marker_colr	SET POLYMARKER COLOUR
pset_marker_rep_plus	SET POLYMARKER REPRESENTATION PLUS
pset_of_fill_area_sets_data	SET OF FILL AREA SETS WITH DATA
pset_of_fill_area_sets3_data	SET OF FILL AREA SETS 3 WITH DATA
pset_param_surf_chars	SET PARAMETRIC SURFACE CHARACTERISTICS
pset_param_surf_ind	SET PARAMETRIC SURFACE INDEX
pset_param_surf_rep	SET PARAMETRIC SURFACE REPRESENTATION
pset_pat_rep_plus	SET PATTERN REPRESENTATION PLUS
pset_refl_ind	SET REFLECTANCE INDEX
pset_refl_model	SET REFLECTANCE MODEL
pset_refl_props	SET REFLECTANCE PROPERTIES
pset_refl_rep	SET REFLECTANCE REPRESENTATION
pset_rend_colr_model	SET RENDERING COLOUR MODEL
pset_surf_approx_crit	SET SURFACE APPROXIMATION CRITERIA
pset_text_colr	SET TEXT COLOUR
pset_text_rep_plus	SET TEXT REPRESENTATION PLUS
ptri_set_data	TRIANGLE SET WITH DATA
ptri_set3_data	TRIANGLE SET 3 WITH DATA
ptri_strip_data	TRIANGLE STRIP WITH DATA
ptri_strip3_data	TRIANGLE STRIP 3 WITH DATA

4.3.2 List ordered alphabetically by PHIGS function name

In table 3 change the clause heading to **List ordered alphabetically by PHIGS and PHIGS PLUS function name**, *change the table caption to* **Table 3 - Function names ordered by PHIGS and PHIGS PLUS names,**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

change the column heading **PHIGS Name** *to* **PHIGS and PHIGS PLUS Name** *and merge alphabetically by function name the following function names:*

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
CELL ARRAY 3 PLUS	pcell_array3_plus
FILL AREA SET WITH DATA	pfill_area_set_data
FILL AREA SET 3 WITH DATA	pfill_area_set3_data
INQUIRE B-SPLINE SURFACE FACILITIES	pinq_b_spline_surf_facs
INQUIRE COLOUR MAPPING FACILITIES	pinq_colr_map_facs
INQUIRE COLOUR MAPPING METHOD FACILITIES	pinq_colr_map_method_facs
INQUIRE COLOUR MAPPING REPRESENTATION	pinq_colr_map_rep
INQUIRE COLOUR MAPPING STATE	pinq_colr_map_st
INQUIRE CURVE FACILITIES	pinq_curve_facs
INQUIRE DATA MAPPING FACILITIES	pinq_data_map_facs
INQUIRE DATA MAPPING REPRESENTATION	pinq_data_map_rep
INQUIRE DEPTH CUE FACILITIES	pinq_depth_cue_facs
INQUIRE DEPTH CUE REPRESENTATION	pinq_depth_cue_rep
INQUIRE DIRECT COLOUR MODEL FACILITIES	pinq_direct_colr_model_facs
INQUIRE DYNAMICS OF WORKSTATION PLUS	pinq_dyns_ws_plus
INQUIRE EDGE REPRESENTATION PLUS	pinq_edge_rep_plus
INQUIRE INTERIOR FACILITIES PLUS	pinq_int_facs_plus
INQUIRE INTERIOR REPRESENTATION PLUS	pinq_int_rep_plus
INQUIRE LIGHT SOURCE FACILITIES	pinq_light_source_facs
INQUIRE LIGHT SOURCE REPRESENTATION	pinq_light_source_rep
INQUIRE LIST OF COLOUR MAPPING INDICES	pinq_list_colr_map_inds
INQUIRE LIST OF DATA MAPPING INDICES	pinq_list_data_map_inds
INQUIRE LIST OF DEPTH CUE INDICES	pinq_list_depth_cue_inds
INQUIRE LIST OF LIGHT SOURCE INDICES	pinq_list_light_source_inds
INQUIRE LIST OF PARAMETRIC SURFACE INDICES	pinq_list_param_surf_inds
INQUIRE LIST OF REFLECTANCE INDICES	pinq_list_refl_inds
INQUIRE PARAMETRIC SURFACE FACILITIES	pinq_param_surf_facs
INQUIRE PARAMETRIC SURFACE REPRESENTATION	pinq_param_surf_rep
INQUIRE PATTERN REPRESENTATION PLUS	pinq_pat_rep_plus
INQUIRE POLYLINE FACILITIES PLUS	pinq_line_facs_plus
INQUIRE POLYLINE REPRESENTATION PLUS	pinq_line_rep_plus
INQUIRE POLYMARKER REPRESENTATION PLUS	pinq_marker_rep_plus
INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION	pinq_pred_colr_map_rep
INQUIRE PREDEFINED DATA MAPPING REPRESENTATION	pinq_pred_data_map_rep
INQUIRE PREDEFINED DEPTH CUE REPRESENTATION	pinq_pred_depth_cue_rep
INQUIRE PREDEFINED EDGE REPRESENTATION PLUS	pinq_pred_edge_rep_plus
INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS	pinq_pred_int_rep_plus
INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION	pinq_pred_light_source_rep
INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION	pinq_pred_param_surf_rep
INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS	pinq_pred_pat_rep_plus
INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS	pinq_pred_line_rep_plus
INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS	pinq_pred_marker_rep_plus
INQUIRE PREDEFINED REFLECTANCE REPRESENTATION	pinq_pred_refl_rep
INQUIRE PREDEFINED TEXT REPRESENTATION PLUS	pinq_pred_text_rep_plus

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
INQUIRE REFLECTANCE FACILITIES	pinq_refl_facs
INQUIRE REFLECTANCE REPRESENTATION	pinq_refl_rep
INQUIRE RENDERING COLOUR MODEL FACILITIES	pinq_rend_colr_model_facs
INQUIRE TEXT REPRESENTATION PLUS	pinq_text_rep_plus
INQUIRE TRIMMING CURVE FACILITIES	pinq_trim_curve_facs
INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS	pinq_ws_st_table_length_plus
NON-UNIFORM B-SPLINE CURVE 3	pnon_uniform_b_spline_curve3
NON-UNIFORM B-SPLINE SURFACE 3	pnon_uniform_b_spline_surf3
NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA	pnon_uniform_b_spline_surf3_data
NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR	pnon_uniform_b_spline_curve3_colr
POLYLINE SET 3 WITH COLOUR	ppolyline_set3_colr
QUADRILATERAL MESH WITH DATA	pquad_mesh_data
QUADRILATERAL MESH 3 WITH DATA	pquad_mesh3_data
SET BACK DATA MAPPING INDEX	pset_back_data_map_ind
SET BACK DATA MAPPING METHOD	pset_back_data_map_method
SET BACK INTERIOR COLOUR	pset_back_int_colr
SET BACK INTERIOR INDEX	pset_back_int_ind
SET BACK INTERIOR SHADING METHOD	pset_back_int_shad_method
SET BACK INTERIOR STYLE INDEX	pset_back_int_style_ind
SET BACK INTERIOR STYLE	pset_back_int_style
SET BACK REFLECTANCE INDEX	pset_back_refl_ind
SET BACK REFLECTANCE MODEL	pset_back_refl_model
SET BACK REFLECTANCE PROPERTIES	pset_back_refl_props
SET COLOUR MAPPING INDEX	pset_colr_map_ind
SET COLOUR MAPPING REPRESENTATION	pset_colr_map_rep
SET CURVE APPROXIMATION CRITERIA	pset_curve_approx_crit
SET DATA MAPPING INDEX	pset_data_map_ind
SET DATA MAPPING METHOD	pset_data_map_method
SET DATA MAPPING REPRESENTATION	pset_data_map_rep
SET DEPTH CUE INDEX	pset_depth_cue_ind
SET DEPTH CUE REPRESENTATION	pset_depth_cue_rep
SET EDGE COLOUR	pset_edge_colr
SET EDGE REPRESENTATION PLUS	pset_edge_rep_plus
SET FACET CULLING MODE	pset_facet_cull_mode
SET FACET DISTINGUISHING MODE	pset_facet_disting_mode
SET INTERIOR COLOUR	pset_int_colr
SET INTERIOR REPRESENTATION PLUS	pset_int_rep_plus
SET INTERIOR SHADING METHOD	pset_int_shad_method
SET LIGHT SOURCE REPRESENTATION	pset_light_source_rep
SET LIGHT SOURCE STATE	pset_light_source_st
SET OF FILL AREA SETS WITH DATA	pset_of_fill_area_sets_data
SET OF FILL AREA SETS 3 WITH DATA	pset_of_fill_area_sets3_data
SET PARAMETRIC SURFACE CHARACTERISTICS	pset_param_surf_chars
SET PARAMETRIC SURFACE INDEX	pset_param_surf_ind
SET PARAMETRIC SURFACE REPRESENTATION	pset_param_surf_rep
SET PATTERN REPRESENTATION PLUS	pset_pat_rep_plus
SET POLYLINE COLOUR	pset_line_colr

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS and PHIGS PLUS function name

PHIGS and PHIGS PLUS Name	C Name
SET POLYLINE REPRESENTATION PLUS	pset_line_rep_plus
SET POLYLINE SHADING METHOD	pset_line_shad_method
SET POLYMARKER COLOUR	pset_marker_colr
SET POLYMARKER REPRESENTATION PLUS	pset_marker_rep_plus
SET REFLECTANCE INDEX	pset_refl_ind
SET REFLECTANCE MODEL	pset_refl_model
SET REFLECTANCE PROPERTIES	pset_refl_props
SET REFLECTANCE REPRESENTATION	pset_refl_rep
SET RENDERING COLOUR MODEL	pset_rend_colr_model
SET SURFACE APPROXIMATION CRITERIA	pset_surf_approx_crit
SET TEXT COLOUR	pset_text_colr
SET TEXT REPRESENTATION PLUS	pset_text_rep_plus
TRIANGLE SET WITH DATA	ptri_set_data
TRIANGLE SET 3 WITH DATA	ptri_set3_data
TRIANGLE STRIP WITH DATA	ptri_strip_data
TRIANGLE STRIP 3 WITH DATA	ptri_strip3_data

--`,,`--`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

5 Type definitions

Change the clause heading to **C PHIGS type definitions.**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

6 Macro definitions

Change the clause heading to **C PHIGS macro definitions.**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7 C PHIGS functions

Change the following function definitions:

INQUIRE GENERALIZED DRAWING PRIMITIVE 3

```
void pinq_gdp3 (

  Pint                  ws_type,        /* workstation type              */
  Pint                  gdp,            /* GDP function identifier       */
  Pint                  *err_ind,       /* OUT error indicator           */
  Pint                  *num_attr,      /* OUT number of attributes used */
  Pattrs                attr[7]         /* OUT list of attributes used   */

);
```

INQUIRE GENERALIZED DRAWING PRIMITIVE

```
void pinq_gdp (

  Pint                  ws_type,        /* workstation type              */
  Pint                  gdp,            /* GDP function identifier       */
  Pint                  *err_ind,       /* OUT error indicator           */
  Pint                  *num_attr,      /* OUT number of attributes used */
  Pattrs                attr[7]         /* OUT list of attributes used   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8 C PHIGS PLUS type definitions

8.1 Mapping of PHIGS PLUS data types

ISO/IEC 9592-4 specifies a set of abstract data types beyond the types defined in ISO/IEC 9592-1. Table 4 - gives the mapping from those additional data types defined in ISO/IEC 9592-4 to the data types defined later in this binding.

Table 4 - Mapping of PHIGS PLUS data types to C

PHIGS PLUS data type		C binding data type
COLRV	colour value	Pcolrv
COLRVH	homogeneous colour value	Phomo_colr_rep_ptr
GCOLR	general colour	Pgcolr
NORM	normal vector	Pvec3
P3H	3D homogeneous point	Phomo_point3
P4H	4D homogeneous point	Phomo_point4

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8.2 Modifications to PHIGS data types

Paspect **ASPECT**

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PASPECT_LINE_COLR,
  PASPECT_MARKER_COLR,
  PASPECT_TEXT_COLR,
  PASPECT_INT_COLR,
  PASPECT_EDGE_COLR,
  PASPECT_LINE_SHAD_METHOD,
  PASPECT_INT_SHAD_METHOD,
  PASPECT_DATA_MAP_METHOD,
  PASPECT_REFL_PROPS,
  PASPECT_REFL_MODEL,
  PASPECT_BACK_INT_STYLE,
  PASPECT_BACK_INT_STYLE_IND,
  PASPECT_BACK_INT_COLR,
  PASPECT_BACK_INT_SHAD_METHOD,
  PASPECT_BACK_DATA_MAP_METHOD,
  PASPECT_BACK_REFL_PROPS,
  PASPECT_BACK_REFL_MODEL,
  PASPECT_CURVE_APPROX_CRIT,
  PASPECT_SURF_APPROX_CRIT,
  PASPECT_PARAM_SURF_CHARS
  /* end of PHIGS PLUS enumerations */
} Paspect;
```

Pattrs **ATTRIBUTES USED**

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PATTR_REFL,
  PATTR_PARAM_SURF
  /* end of PHIGS PLUS enumerations */
} Pattrs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564122 5T9 ■

Pelem_data **ELEMENT DATA**

```
typedef union {
  /* start of PHIGS element data */
  ...,
  /* end of PHIGS element data */
  /* start of PHIGS PLUS element data */
  struct Pelem_polyline_set3_colr {
    Pint            colr_type;          /* colour type                    */
    Pvertex3_set    vertex_data;        /* list of vertex lists           */
  } polyline_set3_colr;
  struct Pelem_fill_area_set3_data {
    Pint            colr_type;          /* colour type                    */
    Pfacet          *facet_data;        /* facet data                     */
    Pedge_flag_set  *edge_flags;        /* list of edge flag lists        */
    Pvertex3_set    vertex_data;        /* list of vertex lists           */
  } fill_area_set3_data;
  struct Pelem_fill_area_set_data {
    Pint            colr_type;          /* colour type                    */
    Pfacet          *facet_data;        /* facet data                     */
    Pedge_flag_set  *edge_flags;        /* list of edge flag lists        */
    Pvertex_set     vertex_data;        /* list of vertex lists           */
  } fill_area_set_data;
  struct Pelem_cell_array3_plus {
    Pparal          paral;              /* parallelogram P, Q, R          */
    Ppat_rep_plus   pattern;            /* pattern colour array           */
  } cell_array3_plus;
  struct Pelem_set_of_fill_area_sets3_data {
    Pint                colr_type;      /* colour type                    */
    Pset_of_fill_area_sets_data *set_data;/* fill area sets data          */
    Pedge_flag_set_list  *edge_flags;   /* list of edge flag sets         */
    Pvertex3_list        vertex_data;   /* list of vertices               */
    Pcontour_set_list    contours;      /* list of contour sets           */
  } set_of_fill_area_sets3_data;
  struct Pelem_set_of_fill_area_sets_data {
    Pint                colr_type;      /* colour type                    */
    Pset_of_fill_area_sets_data *set_data;/* fill area sets data          */
    Pedge_flag_set_list  *edge_flags;   /* list of edge flag sets         */
    Pvertex_list         vertex_data;   /* list of vertices               */
    Pcontour_set_list    contours;      /* list of contour sets           */
  } set_of_fill_area_sets_data;
  struct Pelem_tri_set3_data {
    Pint                    colr_type;  /* colour type                    */
    Ptri_set_data           *set_data;  /* triangle set data              */
    Pedge_flag_triplet_list *edge_flags; /* list of triangle edge flag    */
                                        /* triplets                       */
    Pvertex3_list           vertex_data; /* list of vertices              */
    Pvertex_ind_triplet_list triplets;  /* list of triangle vertex index  */
                                        /* triplets                       */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
} tri_set3_data;
struct Pelem_tri_set_data {
  Pint                    colr_type;    /* colour type                    */
  Ptri_set_data           *set_data;    /* triangle set data              */
  Pedge_flag_triplet_list *edge_flags;  /* list of triangle edge flag     */
                                        /* triplets                       */
  Pvertex_list            vertex_data;  /* list of vertices               */
  Pvertex_ind_triplet_list triplets;    /* list of triangle vertex index  */
                                        /* triplets                       */
} tri_set_data;
struct Pelem_tri_strip3_data {
  Pint                colr_type;        /* colour type                    */
  Pfacet_list         *facet_data;      /* list of facet data             */
  Pedge_flag_list     *edge_flags;      /* list of edge flags             */
  Pvertex3_list       vertex_data;      /* list of vertex data            */
} tri_strip3_data;
struct Pelem_tri_strip_data {
  Pint                colr_type;        /* colour type                    */
  Pfacet_list         *facet_data;      /* list of facet data             */
  Pedge_flag_list     *edge_flags;      /* list of edge flags             */
  Pvertex_list        vertex_data;      /* list of vertex data            */
} tri_strip_data;
struct Pelem_quad_mesh3_data {
  Pint                colr_type;        /* colour type                    */
  Pfacet_array        *facet_data;      /* array of facet data            */
  Pedge_flag_array    *edge_flags;      /* array of edge flags            */
  Pvertex3_array      vertex_data;      /* array of vertex data           */
} quad_mesh3_data;
struct Pelem_quad_mesh_data {
  Pint                colr_type;        /* colour type                    */
  Pfacet_array        *facet_data;      /* array of facet data            */
  Pedge_flag_array    *edge_flags;      /* array of edge flags            */
  Pvertex_array       vertex_data;      /* array of vertex data           */
} quad_mesh_data;
struct Pelem_non_uniform_b_spline_curve3 {
  Pcurve_geom_spline    geom_spline;    /* geometry spline                */
} non_uniform_b_spline_curve3;
struct Pelem_non_uniform_b_spline_curve3_colr {
  Pcurve_geom_spline    geom_spline;    /* geometry spline                */
  Pcurve_colr_spline    *colr_spline;   /* colour spline                  */
} non_uniform_b_spline_curve3_colr;
struct Pelem_non_uniform_b_spline_surf3 {
  Psurf_geom_spline     geom_spline;    /* geometry spline                */
  Ptrim_curve_loop_list *trim_loops;    /* list of trimming curve loops   */
} non_uniform_b_spline_surf3;
struct Pelem_non_uniform_b_spline_surf3_data {
  Psurf_geom_spline     geom_spline;    /* geometry spline                */
  Ptrim_curve_loop_list *trim_loops;    /* list of trimming curve loops   */
  Psurf_colr_spline     *colr_spline;   /* colour spline                  */
  Psurf_data_spline_list *data_spline;  /* data spline                    */
} non_uniform_b_spline_surf3_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    Pgcolr                colr;              /* general colour                */
    Pfacet_disting_mode   facet_disting_mode;/*facet distinguishing mode      */
    Pfacet_cull_mode      facet_cull_mode;   /* facet culling mode            */
    Pdata_map_rep         data_map_method;   /* data mapping method           */
    Prefl_prop            refl_prop;         /* reflectance property data     */
    struct Pelem_light_source_st {
      Pint_list              act_list;       /* activation list               */
      Pint_list              deact_list;     /* deactivation list             */
    } light_source_st;
    Pcurve_approx_crit  curve_approx_crit;   /* curve approximation criteria  */
                                             /* data record                   */
    Psurf_approx_crit   surf_approx_crit;    /* surface approximation criteria */
                                             /* data record                   */
    Pparam_surf_chars   param_surf_chars;    /* parametric surface            */
                                             /* characteristics data record   */
    /* end of PHIGS PLUS element data */

} Pelem_data;
```

NOTE — A NULL value for the facet_data, edge_flags, set_data, colr_spline, trim_loops, or data_spline fields indicates the corresponding optional data is not defined.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pelem_type **ELEMENT TYPE**

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PELEM_POLYLINE_SET3_COLR,
  PELEM_FILL_AREA_SET3_DATA,
  PELEM_FILL_AREA_SET_DATA,
  PELEM_CELL_ARRAY3_PLUS,
  PELEM_SET_OF_FILL_AREA_SETS3_DATA,
  PELEM_SET_OF_FILL_AREA_SETS_DATA,
  PELEM_TRI_SET3_DATA,
  PELEM_TRI_SET_DATA,
  PELEM_TRI_STRIP3_DATA,
  PELEM_TRI_STRIP_DATA,
  PELEM_QUAD_MESH3_DATA,
  PELEM_QUAD_MESH_DATA,
  PELEM_NON_UNIFORM_B_SPLINE_CURVE3,
  PELEM_NON_UNIFORM_B_SPLINE_CURVE3_COLR,
  PELEM_NON_UNIFORM_B_SPLINE_SURF3,
  PELEM_NON_UNIFORM_B_SPLINE_SURF3_DATA,
  PELEM_DATA_MAP_IND,
  PELEM_REFL_IND,
  PELEM_BACK_INT_IND,
  PELEM_BACK_DATA_MAP_IND,
  PELEM_BACK_REFL_IND,
  PELEM_PARAM_SURF_IND,
  PELEM_LINE_COLR,
  PELEM_LINE_SHAD_METHOD,
  PELEM_MARKER_COLR,
  PELEM_TEXT_COLR,
  PELEM_FACET_DISTING_MODE,
  PELEM_FACET_CULL_MODE,
  PELEM_INT_COLR,
  PELEM_INT_SHAD_METHOD,
  PELEM_DATA_MAP_METHOD,
  PELEM_REFL_PROPS,
  PELEM_REFL_MODEL,
  PELEM_BACK_INT_STYLE,
  PELEM_BACK_INT_STYLE_IND,
  PELEM_BACK_INT_COLR,
  PELEM_BACK_INT_SHAD_METHOD,
  PELEM_BACK_DATA_MAP_METHOD,
  PELEM_BACK_REFL_PROPS,
  PELEM_BACK_REFL_MODEL,
  PELEM_LIGHT_SOURCE_ST,
  PELEM_EDGE_COLR,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564126 144 ■


```
    PELEM_CURVE_APPROX_CRIT,
    PELEM_SURF_APPROX_CRIT,
    PELEM_PARAM_SURF_CHARS,
    PELEM_REND_COLR_MODEL,
    PELEM_DEPTH_CUE_IND,
    PELEM_COLR_MAP_IND
    /* end of PHIGS PLUS enumerations */

} Pelem_type;
```

8.3 Implementation dependent PHIGS PLUS type definitions

The following type definitions may be modified by the implementation. If the implementation adds new fields to the unions, the convention specified in clause 5.3 shall be used for naming the new field with the following addition:

For the name of a new field in a PHIGS PLUS implementation dependent type, called "xxx" in clause 5.3, use "type."

`Pcolr_map_method` **COLOUR MAPPING METHOD**

```
typedef struct {

  Pint        method;                    /* colour mapping method          */
  union Pcolr_map_method_data {
    Pint      num_true_colrs;            /* number of true colours         */
    Pint      max_num_pseudo_colrs;      /* maximum number of pseudo       */
                                         /* colours                        */
    ...                                  /* implementation defined         */
  } data;

} Pcolr_map_method;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pcolr_map_rep` **COLOUR MAPPING REPRESENTATION**

```
typedef struct {

  Pint              method;              /* colour mapping method          */
  union Pcolr_map_rep_data {
    struct Pcolr_map_rep_pseudo {
      Pint          model;               /* colour model                   */
      Pfloat_list   weight_vector;       /* list of weight vector          */
      Pcolrv_list   colrs;               /* list of colour values          */
    } pseudo;
    struct Pcolr_map_rep_pseudo_N {
      Pint          model;               /* colour model                   */
      Pfloat_set    colrs;               /* list of lists of colour        */
                                         /* coordinates                    */
    } pseudo_N;
  ...                                    /* implementation defined         */
  } data;

} Pcolr_map_rep;
```

`Pcolr_map_st` **COLOUR MAPPING STATE**

```
typedef struct {

  Pint        method;                    /* colour mapping method          */
  union Pcolr_map_st_data {
    Pint      num_true_colrs;            /* number of available true       */
                                         /* colours                        */
    Pint      max_num_pseudo_colrs;      /* maximum number of pseudo       */
                                         /* colour entries                 */
    ...                                  /* implementation defined         */
  } data;

} Pcolr_map_st;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564128 T17 ■

`Pcolr_rep_ptr` **COLOUR REPRESENTATION POINTER**

```
typedef union {

  Prgb              *rgb;              /* pointer to RGB colour values    */
  Pcieluv           *cieluv;           /* pointer to CIELUV colour values*/
  Phls              *hls;              /* pointer to HLS colour values    */
  Phsv              *hsv;              /* pointer to HSV colour values    */
  Pdata             *unsupp;           /* pointer to unsupported colour   */
                                       /* values                          */
  ...                                  /* implementation defined          */

} Pcolr_rep_ptr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pcurve_approx_crit **CURVE APPROXIMATION CRITERIA**

```
typedef struct {
  Pint              type;                 /* curve approximation criteria   */
                                          /* type                           */
  union Pcurve_approx_crit_data {
    struct Pcurve_approx_const_param_subd {
      Pint          count;                /* count                          */
    } const_param_subd;
    struct Pcurve_approx_chordal_size_wc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_wc;
    struct Pcurve_approx_chordal_size_npc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_npc;
    struct Pcurve_approx_chordal_size_dc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_dc;
    struct Pcurve_approx_chordal_dev_wc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_wc;
    struct Pcurve_approx_chordal_dev_npc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_npc;
    struct Pcurve_approx_chordal_dev_dc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_dc;
    struct Pcurve_approx_rel_wc {
      Pfloat        approx;               /* approximation value            */
    } rel_wc;
    struct Pcurve_approx_rel_npc {
      Pfloat        approx;               /* approximation value            */
    } rel_npc;
    struct Pcurve_approx_rel_dc {
      Pfloat        approx;               /* approximation value            */
    } rel_dc;
    Pdata           unsupp;               /* unsupported curve approximation*/
                                          /* criteria data                  */
    ...                                   /* implementation defined         */
  } data;

} Pcurve_approx_crit;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pdata_map_rep **DATA MAPPING REPRESENTATION**

```
typedef struct {

  Pint                    method;         /* data mapping method            */
  Psource_select_list     selectors;      /* list of source selectors       */
  union Pdata_map_rep_data {
    struct Pdata_map_rep_single_uniform {
      Pint                ind;            /* data value index               */
      Pfloat              lower_limit;    /* lower range limit              */
      Pfloat              upper_limit;    /* upper range limit              */
      Pint                colr_type;      /* colour type                    */
      Pcolrv_list         colr_values;    /* list of colour values          */
    } single_uniform;
    struct Pdata_map_rep_single_non_uniform {
      Pint                ind;            /* data value index               */
      Pfloat_list         range;          /* range boundaries               */
      Pint                colr_type;      /* colour type                    */
      Pcolrv_list         colr_values;    /* list of colour values          */
    } single_non_uniform;
    struct Pdata_map_rep_bi_uniform {
      Pint                inds[2];        /* data value indices             */
      Pfloat              lower_limit_a;  /* lower limit of Ra range        */
      Pfloat              upper_limit_a;  /* upper limit of Ra range        */
      Pfloat              lower_limit_b;  /* lower limit of Rb range        */
      Pfloat              upper_limit_b;  /* upper limit of Rb range        */
      Pint                colr_type;      /* colour type                    */
      Pcolrv_set          colr_values;    /* list of colour value lists     */
    } bi_uniform;
    struct Pdata_map_rep_bi_non_uniform {
      Pint                inds[2];        /* data value indices             */
      Pfloat_list         range_a;        /* Ra range boundaries            */
      Pfloat_set          range_b;        /* Rb range boundaries            */
      Pint                colr_type;      /* colour type                    */
      Pcolrv_set          colr_values;    /* array of colour values         */
    } bi_non_uniform;
    ...                                   /* implementation defined         */
  } data;

} Pdata_map_rep;
```

NOTE — The indexing of data mapping values is 1 based.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Phomo_colr_rep_ptr **HOMOGENEOUS COLOUR REPRESENTATION POINTER**

```
typedef union {
  Phomo_rgb         *rgb;          /* pointer to homogeneous rgb     */
                                   /* colour value                   */
  Phomo_cieluv      *cieluv;       /* pointer to homogeneous cieluv  */
                                   /* colour value                   */
  Phomo_hls         *hls;          /* pointer to homogeneous hls     */
                                   /* colour value                   */
  Phomo_hsv         *hsv;          /* pointer to homogeneous hsv     */
                                   /* colour value                   */
  ...                              /* implementation defined         */
} Phomo_colr_rep_ptr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564132 448 ■

Plight_source_rep **LIGHT SOURCE REPRESENTATION**

```
typedef struct {

  Pint              type;               /* light source type            */
  union Plight_source_rep_data {
    struct Plight_source_rep_amb {
      Pgcolr        colr;               /* light source colour          */
    } amb;
    struct Plight_source_rep_dir {
      Pgcolr        colr;               /* light source colour          */
      Pvec3         dir;                /* light source direction       */
    } dir;
    struct Plight_source_rep_pos {
      Pgcolr        colr;               /* light source colour          */
      Ppoint3       pos;                /* light source position        */
      Pfloat        c1;                 /* light source attenuation     */
                                        /* coefficient                  */
      Pfloat        c2;                 /* light source attenuation     */
                                        /* coefficient                  */
    } pos;
    struct Plight_source_rep_spot {
      Pgcolr        colr;               /* light source colour          */
      Ppoint3       pos;                /* light source position        */
      Pvec3         dir;                /* light source direction       */
      Pfloat        concent_exp;        /* concentration exponent       */
      Pfloat        c1;                 /* light source attenuation     */
                                        /* coefficient                  */
      Pfloat        c2;                 /* light source attenuation     */
                                        /* coefficient                  */
      Pfloat        spread_angle;       /* light source spread angle    */
    } spot;
    ...                                 /* implementation defined       */
  } data;

} Plight_source_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pparam_surf_chars **PARAMETRIC SURFACE CHARACTERISTICS**

```
typedef struct {
  Pint              type;              /* parametric surface            */
                                       /* characteristics type          */
  union Pparam_surf_chars_data {
    struct Pparam_surf_isoparam {
      Pplacement    placement;         /* curve placement: UNIFORM OVER */
                                       /* SURFACE, UNIFORM BETWEEN KNOTS */
      Pint          num_u_curves;      /* curve count in u direction    */
      Pint          num_v_curves;      /* curve count in v direction    */
    } isoparam;
    struct Pparam_surf_level_mc {
      Ppoint3       origin;            /* origin point                  */
      Pvec3         dir;               /* direction vector              */
      Pfloat_list   params;            /* parameter list                */
    } level_mc;
    struct Pparam_surf_level_wc {
      Ppoint3       origin;            /* origin point                  */
      Pvec3         dir;               /* direction vector              */
      Pfloat_list   params;            /* parameter list                */
    } level_wc;
    Pdata           unsupp;            /* unsupported parametric        */
                                       /* surface characteristics data  */
    ...                                /* implementation defined        */
  } data;

} Pparam_surf_chars;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Prefl_prop **REFLECTANCE PROPERTY**

```
typedef struct {
  Pint              type;              /* reflectance property type      */
  union Prefl_prop_data {
    struct Prefl_prop_simple_refl {
      Pfloat        amb_coeff;         /* ambient reflection             */
                                       /* coefficient                    */
      Pfloat        diff_coeff;        /* diffuse reflection             */
                                       /* coefficient                    */
      Pfloat        spec_coeff;        /* specular reflection            */
                                       /* coefficient                    */
      Pgcolr        spec_colr;         /* specular colour                */
      Pfloat        spec_exp;          /* specular exponent              */
    } simple_refl;
    Pdata           unsupp;            /* unsupported reflectance        */
                                       /* property data                  */
    ...                                /* implementation defined         */
  } data;
} Prefl_prop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Psurf_approx_crit **SURFACE APPROXIMATION CRITERIA**

```
typedef struct {
  Pint              type;               /* surface approximation criteria */
                                        /* type                           */
  union Psurf_approx_crit_data {
    struct Psurf_approx_const_param_subd {
      Pint          u_count;            /* u count                        */
      Pint          v_count;            /* v count                        */
    } const_param_subd;
    struct Psurf_approx_chordal_size_wc {
      Pfloat        u_approx;           /* u approximation value          */
      Pfloat        v_approx;           /* v approximation value          */
    } chordal_size_wc;
    struct Psurf_approx_chordal_size_npc {
      Pfloat        u_approx;           /* u approximation value          */
      Pfloat        v_approx;           /* v approximation value          */
    } chordal_size_npc;
    struct Psurf_approx_chordal_aize_dc {
      Pfloat        u_approx;           /* u approximation value          */
      Pfloat        v_approx;           /* v approximation value          */
    } chordal_size_dc;
    struct Psurf_approx_planar_dev_wc {
      Pfloat        approx;             /* approximation value            */
    } planar_dev_wc;
    struct Psurf_approx_planar_dev_npc {
      Pfloat        approx;             /* approximation value            */
    } planar_dev_npc;
    struct Psurf_approx_planar_dev_dc {
      Pfloat        approx;             /* approximation value            */
    } planar_dev_dc;
    struct Psurf_approx_rel_wc {
      Pfloat        approx;             /* approximation value            */
    } rel_wc;
    struct Psurf_approx_rel_npc {
      Pfloat        approx;             /* approximation value            */
    } rel_npc;
    struct Psurf_approx_rel_dc {
      Pfloat        approx;             /* approximation value            */
    } rel_dc;
    Pdata           unsupp;             /* unsupported surface            */
                                        /* approximation criteria data    */
    ...                                 /* implementation defined         */
  } data;

} Psurf_approx_crit;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564136 093 ■

8.4 Implementation independent PHIGS PLUS type definitions

`Pcolr_ctrl_point_array` **COLOUR CONTROL POINT ARRAY**

```
typedef struct {
  Prationality          rationality;    /* control point rationality      */
  Pint_size             dims;           /* dimensions of control point    */
                                        /* array                          */
  Pcolr_ctrl_point_ptr  colr_points;    /* array of colour control points */
} Pcolr_ctrl_point_array;
```

`Pcolr_ctrl_point_list` **COLOUR CONTROL POINT LIST**

```
typedef struct {
  Prationality          rationality;    /* control point rationality      */
  Pint                  num_points;     /* number of control points       */
  Pcolr_ctrl_point_ptr  points;         /* list of colour control points  */
} Pcolr_ctrl_point_list;
```

`Pcolr_ctrl_point_ptr` **COLOUR CONTROL POINT POINTER**

```
typedef union {
  Pcolr_rep_ptr         points;         /* pointer to colour points       */
  Phomo_colr_rep_ptr    homo_points;    /* pointer to homogeneous colour  */
                                        /* points                         */
} Pcolr_ctrl_point_ptr;
```

`Pcolrv` **COLOUR VALUE**

```
typedef union {
  Pint                  colr_ind;       /* colour index                   */
  Pcolr_rep             colr_rep;       /* colour representation          */
} Pcolrv;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Pcolrv_array` **COLOUR VALUE ARRAY**

```
typedef struct {
  Pint_size         dims;              /* colour value array dimensions  */
  Pcolrv_ptr        colrs;             /* array of colour values         */
} Pcolrv_array;
```

`Pcolrv_list` **COLOUR VALUE LIST**

```
typedef struct {
  Pint              num_colrs;         /* number of colours              */
  Pcolrv_ptr        colrs;             /* list of colour values          */
} Pcolrv_list;
```

`Pcolrv_ptr` **COLOUR VALUE POINTER**

```
typedef union {
  Pint              *colr_inds;        /* pointer to colour indices      */
  Pcolr_rep_ptr     colr_reps;         /* pointer to colour              */
                                       /* representations                */
} Pcolrv_ptr;
```

`Pcolrv_set` **COLOUR VALUE SET**

```
typedef struct {
  Pint              num_lists;         /* number of colour value lists   */
  Pcolrv_list       *colrs;            /* list of colour value lists     */
} Pcolrv_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Pcontour_set` **CONTOUR SET**

```
typedef struct {
  Pint              num_lists;          /* number of contour lists        */
  Pint_list         *contours;          /* list of contour lists          */
} Pcontour_set;
```

`Pcontour_set_list` **CONTOUR SET LIST**

```
typedef struct {
  Pint              num_contour_sets;   /* number of contour sets         */
  Pcontour_set      *contours;          /* list of contour sets           */
} Pcontour_set_list;
```

`Pctrl_point3_array` **CONTROL POINT 3 ARRAY**

```
typedef struct {
  Prationality      rationality;        /* control point rationality      */
  Pint_size         dims;               /* control point array u,v        */
                                        /* dimensions                     */
  Ppoint3_ptr       points;             /* array of control points        */
} Pctrl_point3_array;
```

`Pctrl_point_list` **CONTROL POINT LIST**

```
typedef struct {
  Prationality      rationality;        /* control point rationality      */
  Pint              num_points;         /* number of control points       */
  Ppoint_ptr        points;             /* list of control points         */
} Pctrl_point_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pctrl_point3_list` **CONTROL POINT 3 LIST**

```
typedef struct {
  Prationality      rationality;        /* control point rationality      */
  Pint              num_points;         /* number of control points       */
  Ppoint3_ptr       points;             /* list of control points         */
} Pctrl_point3_list;
```

`Pcurve_colr_spline` **CURVE COLOUR SPLINE**

```
typedef struct {
  Pint                    order;        /* spline order                   */
  Pfloat_list             knots;        /* list of knot values            */
  Pint                    colr_type;    /* colour type                    */
  Pcolr_ctrl_point_list   colrs;        /* list of colour control points  */
} Pcurve_colr_spline;
```

`Pcurve_geom_spline` **CURVE GEOMETRY SPLINE**

```
typedef struct {
  Pint                  order;          /* spline order                   */
  Pfloat_list           knots;          /* knots                          */
  Pfloat                low_limit;      /* lower parameter range limit    */
  Pfloat                high_limit;     /* high parameter range limit     */
  Pctrl_point3_list     ctrl_points;    /* list of control points         */
} Pcurve_geom_spline;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pdata_ctrl_point_array **DATA CONTROL POINT ARRAY**

```
typedef struct {

  Prationality      rationality;        /* control point rationality      */
  Pint_size         dims;               /* control point array u,v        */
                                        /* dimensions                     */
  Pint              points_dims;        /* control point dimensionality   */
  Pfloat            *ctrl_points;       /* array of control points        */

} Pdata_ctrl_point_array;
```

Pdepth_cue_mode **DEPTH CUE MODE**

```
typedef enum {

  PDEPTH_CUE_SUPPR,
  PDEPTH_CUE_ALLOWED

} Pdepth_cue_mode;
```

Pdepth_cue_mode_list **DEPTH CUE MODE LIST**

```
typedef struct {

  Pint              num_modes;          /* number of depth cue modes      */
  Pdepth_cue_mode   *modes;             /* list of depth cue modes        */

} Pdepth_cue_mode_list;
```

Pdepth_cue_rep **DEPTH CUE REPRESENTATION**

```
typedef struct {

  Pdepth_cue_mode   mode;               /* depth cue mode                 */
  Pfloat            ref_planes[2];      /* depth cue reference planes     */
                                        /* (min, max)
  Pfloat            scale[2];           /* depth cue scale factors        */
                                        /* (min, max)
  Pgcolr            gcolr;              /* depth cue colour               */

} Pdepth_cue_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pdyn_ws_attrs_plus` **DYNAMICS OF WORKSTATION ATTRIBUTES PLUS**

```
typedef struct {

  Pdyn_mod          data_map;          /* data mapping representation    */
  Pdyn_mod          refl;              /* reflectance representation     */
  Pdyn_mod          param_surf;        /* parametric surface             */
                                       /* representation                 */
  Pdyn_mod          light_source;      /* light source representation    */
  Pdyn_mod          depth_cue;         /* depth cue representation       */
  Pdyn_mod          colr_map;          /* colour mapping representation  */

} Pdyn_ws_attrs_plus;
```

`Pedge_bundle_plus` **EDGE BUNDLE PLUS**

```
typedef struct {

  Pedge_flag        flag;              /* edge flag                      */
  Pint              type;              /* edgetype                       */
  Pfloat            width;             /* edgewidth scale factor         */
  Pgcolr            colr;              /* edge colour                    */

} Pedge_bundle_plus;
```

`Pedge_flag_array` **EDGE FLAG ARRAY**

```
typedef struct {

  Pint_size         dims;              /* edge flag array dimensions     */
  Pedge_flag_pair   *edge_flags;       /* array of edge flag pairs       */

} Pedge_flag_array;
```

`Pedge_flag_list` **EDGE FLAG LIST**

```
typedef struct {

  Pint              num_edges;         /* number of edges in list        */
  Pedge_flag        *edge_flags;       /* list of edge flags             */

} Pedge_flag_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pedge_flag_pair` **EDGE FLAG PAIR**

```
typedef Pedge_flag Pedge_flag_pair[2];
```

`Pedge_flag_set` **EDGE FLAG SET**

```
typedef struct {
  Pint              num_lists;          /* number of edge flag lists      */
  Pedge_flag_list   *edge_flags;        /* list of edge flag lists        */
} Pedge_flag_set;
```

`Pedge_flag_set_list` **EDGE FLAG SET LIST**

```
typedef struct {
  Pint              num_edge_flag_sets; /* number of edge flag sets       */
  Pedge_flag_set    *edge_flag_sets;    /* list of edge flag sets         */
} Pedge_flag_set_list;
```

`Pedge_flag_triplet` **EDGE FLAG TRIPLET**

```
typedef Pedge_flag Pedge_flag_triplet[3];
```

`Pedge_flag_triplet_list` **EDGE FLAG TRIPLET LIST**

```
typedef struct {
  Pint              num_edge_flag_triplets;/*number of edge flag triplets  */
  Pedge_flag_triplet*edge_flag_triplets; /* list of edge flag triplets     */
} Pedge_flag_triplet_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pfacet` **FACET**

```
typedef struct {

  Pint              num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv            *facet_colrv;        /* facet colour                   */
  Pvec3             *facet_norm;         /* facet normal                   */
  Pfloat            *facet_data;         /* list of facet data             */

} Pfacet;
```

NOTE — A NULL value for the `facet_colrv`, `facet_norm`, or `facet_data` fields indicates the corresponding optional data is not supplied.

`Pfacet_array` **FACET ARRAY**

```
typedef struct {

  Pint_size         dims;                /* facet array dimensions         */
  Pint              num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv_ptr        facet_colrvs;        /* array of facet colour values   */
  Pvec3             *facet_norms;        /* array of facet normals         */
  Pfloat            *facet_data;         /* array of facet data lists      */

} Pfacet_array;
```

NOTE — A NULL value for the `facet_colrvs`, `facet_norms`, or `facet_data` fields indicates the corresponding optional data is not supplied.

`Pfacet_cull_mode` **FACET CULLING MODE**

```
typedef enum {

  PFACET_CULL_NONE,
  PFACET_CULL_BACKFACING,
  PFACET_CULL_FRONTFACING

} Pfacet_cull_mode;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pfacet_disting_mode` **FACET DISTINGUISHING MODE**

```
typedef enum {
  PFACET_DISTING_OFF,
  PFACET_DISTING_ON
} Pfacet_disting_mode;
```

`Pfacet_list` **FACET LIST**

```
typedef struct {
  Pint          num_facets;          /* number of facets                */
  Pint          num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv_ptr    facet_colrvs;        /* list of facet colour values     */
  Pvec3         *facet_norms;        /* list of facet normals           */
  Pfloat        *facet_data;         /* list of facet data lists        */
} Pfacet_list;
```

NOTE — A NULL value for the `facet_colrvs`, `facet_norms`, or `facet_data` fields indicates the corresponding optional data is not supplied.

`Pfloat_list` **FLOAT LIST**

```
typedef struct {
  Pint          num_floats;          /* number of floats in list        */
  Pfloat        *floats;             /* list of floats                  */
} Pfloat_list;
```

`Pfloat_set` **FLOAT SET**

```
typedef struct {
  Pint          num_lists;           /* number of float lists           */
  Pfloat_list   *floats;             /* list of float lists             */
} Pfloat_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pgcolr` **GENERAL COLOUR**

```
typedef struct {
  Pint              colr_type;          /* colour type                */
  Pcolrv            colr_value;         /* colour value               */
} Pgcolr;
```

`Phomo_cieluv` **HOMOGENEOUS CIEL*U*V***

```
typedef struct {
  Pfloat            cieluv_x;           /* x coefficient              */
  Pfloat            cieluv_y;           /* y coefficient              */
  Pfloat            cieluv_y_lum;       /* y luminance                */
  Pfloat            w;                  /* homogeneous coordinate     */
} Phomo_cieluv;
```

`Phomo_hls` **HOMOGENEOUS HUE LIGHTNESS SATURATION**

```
typedef struct {
  Pfloat            hue;                /* hue                        */
  Pfloat            lightness;          /* lightness                  */
  Pfloat            satur;              /* saturation                 */
  Pfloat            w;                  /* homogeneous coordinate     */
} Phomo_hls;
```

`Phomo_hsv` **HOMOGENEOUS HUE SATURATION VALUE**

```
typedef struct {
  Pfloat            hue;                /* hue                        */
  Pfloat            satur;              /* saturation                 */
  Pfloat            value;              /* value                      */
  Pfloat            w;                  /* homogeneous coordinate     */
} Phomo_hsv;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564146 T32 ■

Phomo_point3 **HOMOGENEOUS POINT 3**

```
typedef struct {
  Pfloat          wx;             /* wx coordinate                */
  Pfloat          wy;             /* wy coordinate                */
  Pfloat          w;              /* homogeneous coordinate       */
} Phomo_point3;
```

Phomo_point4 **HOMOGENEOUS POINT 4**

```
typedef struct {
  Pfloat          wx;             /* wx coordinate                */
  Pfloat          wy;             /* wy coordinate                */
  Pfloat          wz;             /* wz coordinate                */
  Pfloat          w;              /* homogeneous coordinate       */
} Phomo_point4;
```

Phomo_rgb **HOMOGENEOUS RED GREEN BLUE**

```
typedef struct {
  Pfloat          red;            /* red                          */
  Pfloat          green;          /* green                        */
  Pfloat          blue;           /* blue                         */
  Pfloat          w;              /* homogeneous coordinate       */
} Phomo_rgb;
```

Pint_bundle_plus **INTERIOR BUNDLE PLUS**

```
typedef struct {
  Pint_style      style;          /* interior style               */
  Pint            style_ind;      /* interior style index         */
  Pgcolr          colr;           /* interior colour              */
  Pint            shad_method;    /* shading method               */
} Pint_bundle_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pint_facs_plus` **INTERIOR FACILITIES PLUS**

```
typedef struct {
  Pint              num_int_styles;      /* number of interior styles      */
  Pint_style        int_styles[5];       /* list of available interior     */
                                         /* styles                         */
  Pint              num_hatch_styles;    /* number of available hatch      */
                                         /* styles                         */
  Pint_list         hatch_styles;        /* list of available hatch styles */
  Pint_list         shad_methods;        /* list of available shading      */
                                         /* methods                        */
  Pint              num_pred_int_inds;   /* number of predefined interior  */
                                         /* indices                        */

} Pint_facs_plus;
```

`Pline_bundle_plus` **POLYLINE BUNDLE PLUS**

```
typedef struct {
  Pint                type;              /* line type                      */
  Pfloat              width;             /* linewidth scale factor         */
  Pgcolr              colr;              /* polyline colour                */
  Pint                shad_method;       /* polyline shading method        */
  Pcurve_approx_crit  approx_crit;       /* curve approximation criteria   */
                                         /* data record                    */

} Pline_bundle_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pline_facs_plus` **POLYLINE FACILITIES PLUS**

```
typedef struct {

  Pint              num_types;          /* number of available linetypes  */
  Pint_list         types;              /* list of line types             */
  Pint              num_widths;         /* number of available line widths*/
  Pfloat            nom_width;          /* nominal linewidth              */
  Pfloat            min_width;          /* minimum linewidth              */
  Pfloat            max_width;          /* maximum linewidth              */
  Pint_list         shad_methods;       /* list of available line shading */
                                        /* methods                        */
  Pint              num_inds;           /* number of predefined bundle    */
                                        /* indices                        */

} Pline_facs_plus;
```

`Pmarker_bundle_plus` **POLYMARKER BUNDLE PLUS**

```
typedef struct {

  Pint              type;               /* marker type                    */
  Pfloat            size;               /* marker size scale factor       */
  Pgcolr            colr;               /* polymarker colour              */

} Pmarker_bundle_plus;
```

`Pparam_surf_rep` **PARAMETRIC SURFACE REPRESENTATION**

```
typedef struct {

  Psurf_approx_crit     approx_crit;    /* surface approximation criteria */
                                        /* data                           */
  Pparam_surf_chars     chars_data;     /* parametric surface             */
                                        /* characteristics data           */

} Pparam_surf_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Ppat_rep_plus` **PATTERN REPRESENTATION PLUS**

```
typedef struct {

  Pint                colr_type;      /* colour type                  */
  Pcolrv_array        colrs;          /* array of colour values       */

} Ppat_rep_plus;
```

`Pplacement` **PLACEMENT**

```
typedef enum {

  PPLACE_UNIFORM_OVER_SURF,
  PPLACE_UNIFORM_BETWEEN_KNOTS

} Pplacement;
```

`Ppoint_ptr` **POINT POINTER**

```
typedef union {

  Ppoint          *point;           /* pointer to 2D non-homogeneous  */
                                    /* points                         */
  Phomo_point3    *homo_point3;     /* pointer to 3D homogeneous      */
                                    /* points                         */

} Ppoint_ptr;
```

`Ppoint3_ptr` **POINT 3 POINTER**

```
typedef union {

  Ppoint3         *point3;          /* pointer to 3D non-homogeneous  */
                                    /* points                         */
  Phomo_point4    *homo_point4;     /* pointer to 4D homogeneous      */
                                    /* points                         */

} Ppoint3_ptr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Prationality` **RATIONALITY**

```
typedef enum {
  PB_SPLINE_RATIONAL,
  PB_SPLINE_NON_RATIONAL
} Prationality;
```

`Prefl_rep` **REFLECTANCE REPRESENTATION**

```
typedef struct {
  Pint              refl_model;       /* reflectance model              */
  Prefl_prop        refl_prop;        /* reflectance property data      */
} Prefl_rep;
```

`Pset_of_fill_area_sets_data` **SET OF FILL AREA SETS DATA**

```
typedef struct {
  Pint              num_sets;         /* number of fill area sets       */
  Pint              num_data_per_facet; /* number of data values per facet*/
  Pcolrv_ptr        facet_colrvs;     /* list of facet colours          */
  Pvec3             *facet_norms;     /* list of facet normals          */
  Pfloat            *facet_data;      /* list of facet data lists       */
} Pset_of_fill_area_sets_data;
```

NOTE — A NULL value for the `facet_colrvs`, `facet_norms`, or `facet_data` fields indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Psource_select` **SOURCE SELECT**

```
typedef enum {
  PSOURCE_SELECT_COLR_ASPECT,
  PSOURCE_SELECT_VERT_COLR,
  PSOURCE_SELECT_VERT_DATA,
  PSOURCE_SELECT_FACET_COLR,
  PSOURCE_SELECT_FACET_DATA
} Psource_select;
```

`Psource_select_list` **SOURCE SELECT LIST**

```
typedef struct {
  Pint              num_selectors;      /* number of selectors          */
  Psource_select    *selectors;         /* list of source selectors     */
} Psource_select_list;
```

`Psurf_colr_spline` **SURFACE COLOUR SPLINE**

```
typedef struct {
  Pint                      u_order;     /* u spline order                */
  Pint                      v_order;     /* v spline order                */
  Pfloat_list               u_knots;     /* list of u knot values         */
  Pfloat_list               v_knots;     /* list of v knot values         */
  Pint                      colr_type;   /* colour type                   */
  Pcolr_ctrl_point_array    colrs;       /* array of colour control points */
} Psurf_colr_spline;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Psurf_data_spline` **SURFACE DATA SPLINE**

```
typedef struct {

  Pint                    u_order;      /* u spline order                */
  Pint                    v_order;      /* v spline order                */
  Pfloat_list             u_knots;      /* list of u knot values         */
  Pfloat_list             v_knots;      /* list of v knot values         */
  Pdata_ctrl_point_array  data_points;  /* array of data control         */
                                        /* points                        */

} Psurf_data_spline;
```

`Psurf_data_spline_list` **SURFACE DATA SPLINE LIST**

```
typedef struct {

  Pint                num_data_spline;   /* number of data splines       */
  Psurf_data_spline   *data_splines;     /* list of data splines         */

} Psurf_data_spline_list;
```

`Psurf_geom_spline` **SURFACE GEOMETRY SPLINE**

```
typedef struct {

  Pint                  u_order;        /* u spline order                */
  Pint                  v_order;        /* v spline order                */
  Pfloat_list           u_knots;        /* u knots                       */
  Pfloat_list           v_knots;        /* v knots                       */
  Pctrl_point3_array    ctrl_points;    /* array of control points       */

} Psurf_geom_spline;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Ptext_bundle_plus` **TEXT BUNDLE PLUS**

```
typedef struct {
  Pint              font;              /* text font                    */
  Ptext_prec        precision;         /* text precision               */
  Pfloat            char_expan;        /* character expansion factor   */
  Pfloat            char_space;        /* character spacing            */
  Pgcolr            colr;              /* text colour                  */
} Ptext_bundle_plus;
```

`Ptri_set_data` **TRIANGLE SET DATA**

```
typedef struct {
  Pint              num_sets;          /* number of triangle sets      */
  Pint              num_data_per_facet;/*number of data values per     */
                                       /* facet                        */
  Pcolrv_ptr        facet_colrvs;      /* list of facet colours        */
  Pvec3             *facet_norms;      /* list of facet normals        */
  Pfloat            *facet_data;       /* list of facet data lists     */
} Ptri_set_data;
```

NOTE — A NULL value for the `facet_colrvs`, `facet_norms`, `facet_data`, or `edge_flags` fields indicates the corresponding optional data is not supplied.

`Ptrim_curve` **TRIMMING CURVE**

```
typedef struct {
  Pcurve_approx_crit  approx_crit;     /* trimming curve approx.       */
                                       /* criteria data record         */
  Pvis_flag           visibility_flag; /* curve visibility flag        */
  Pint                order;           /* curve order                  */
  Pfloat_list         knots;           /* curve knot vector            */
  Pfloat              low_limit;       /* curve parameter range lower  */
                                       /* limit                        */
  Pfloat              high_limit;      /* curve parameter range upper  */
                                       /* limit                        */
  Pctrl_point_list    ctrl_points;     /* curve control points         */
} Ptrim_curve;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564154 009 ■

`Ptrim_curve_loop` **TRIMMING CURVE LOOP**

```
typedef struct {

  Pint              num_trim_curves;     /* number of trimming curves      */
  Ptrim_curve       *trim_curves;        /* list of trimming curves forming*/
                                         /* a closed loop                  */

} Ptrim_curve_loop;
```

`Ptrim_curve_loop_list` **TRIMMING CURVE LOOP LIST**

```
typedef struct {

  Pint                 num_trim_loops;  /* number of trimming curve loops */
  Ptrim_curve_loop     *trim_loops;     /* list of trimming curve loops   */

} Ptrim_curve_loop_list;
```

`Pvertex_array` **VERTEX ARRAY**

```
typedef struct {

  Pint_size         dims;                /* dimensions of vertex array     */
  Pint              num_data_per_vertex; /* number of data values          */
                                         /* per-vertex                     */
  Ppoint            *vertex_points;      /* array of vertex points         */
  Pcolrv_ptr        vertex_colrvs;       /* array of vertex colours        */
  Pvec3             *vertex_norms;       /* array of vertex normals        */
  Pfloat            *vertex_data;        /* array of vertex data lists     */

} Pvertex_array;
```

NOTE — A NULL value for the `vertex_colrvs`, `vertex_norms`, or `vertex_data` fields indicates the corresponding optional data is not supplied.

`Pvertex_ind_triplet` **VERTEX INDEX TRIPLET**

```
typedef Pint Pvertex_ind_triplet[3];
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

`Pvertex_ind_triplet_list` **VERTEX INDEX TRIPLET LIST**

```
typedef struct {
  Pint           num_vertex_ind_triplets;/* number of vertex index        */
                                         /* triplets                      */
  Pvertex_ind_triplets*vertex_ind_triplets;/*list of vertex index triplets */
} Pvertex_ind_triplet_list;
```

`Pvertex_list` **VERTEX LIST**

```
typedef struct {
  Pint             num_vertices;        /* number of vertices            */
  Pint             num_data_per_vertex; /* number of data values         */
                                        /* per-vertex                    */
  Ppoint           *vertex_points;      /* list of vertex points         */
  Pcolrv_ptr       vertex_colrvs;       /* list of vertex colours        */
  Pvec3            *vertex_norms;       /* list of vertex normals        */
  Pfloat           *vertex_data;        /* list of vertex data lists     */
} Pvertex_list;
```

NOTE — A NULL value for the `vertex_colrvs`, `vertex_norms`, or `vertex_data` fields indicates the corresponding optional data is not supplied.

`Pvertex_set` **VERTEX SET**

```
typedef struct {
  Pint             num_lists;           /* number of vertex lists        */
  Pvertex_list     *vertices;           /* list of vertex lists          */
} Pvertex_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564156 981 ■

`Pvertex3_array` **VERTEX 3 ARRAY**

```
typedef struct {

  Pint_size         dims;                /* dimensions of vertex array     */
  Pint              num_data_per_vertex; /* number of data values          */
                                         /* per-vertex                     */
  Ppoint3           *vertex_points;      /* array of vertex points         */
  Pcolrv_ptr        vertex_colrvs;       /* array of vertex colours        */
  Pvec3             *vertex_norms;       /* array of vertex normals        */
  Pfloat            *vertex_data;        /* array of vertex data lists     */

} Pvertex3_array;
```

NOTE — A `NULL` value for the `vertex_colrvs`, `vertex_norms`, or `vertex_data` fields indicates the corresponding optional data is not supplied.

`Pvertex3_list` **VERTEX 3 LIST**

```
typedef struct {

  Pint              num_vertices;        /* number of vertices             */
  Pint              num_data_per_vertex; /* number of data values          */
                                         /* per-vertex                     */
  Ppoint3           *vertex_points;      /* list of vertex points          */
  Pcolrv_ptr        vertex_colrvs;       /* list of vertex colours         */
  Pvec3             *vertex_norms;       /* list of vertex normals         */
  Pfloat            *vertex_data;        /* list of vertex data lists      */

} Pvertex3_list;
```

NOTE — A `NULL` value for the `vertex_colrvs`, `vertex_norms`, or `vertex_data` fields indicates the corresponding optional data is not supplied.

`Pvertex3_set` **VERTEX 3 SET**

```
typedef struct {

  Pint              num_lists;           /* number of vertex lists         */
  Pvertex3_list     *vertices;           /* list of vertex lists           */

} Pvertex3_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

`Pvis_flag` **VISIBILITY FLAG**

```
typedef enum {
  PVIS_OFF,
  PVIS_ON
} Pvis_flag;
```

`Pws_st_tables_plus` **LENGTHS OF WORKSTATION STATE TABLES**

```
typedef struct {
  Pint              data_map_rep;       /* max. # of data mapping table   */
                                        /* entries                        */
  Pint              refl_rep;           /* max. # of reflectance table    */
                                        /* entries                        */
  Pint              param_surf_rep;     /* max. # of param. surface table */
                                        /* entries                        */
  Pint              light_source_rep;   /* max. # of light source table   */
                                        /* entries                        */
  Pint              depth_cue_rep;      /* max. # of depth cue table      */
                                        /* entries                        */
  Pint              colr_map_rep;       /* max. # of colour mapping table */
                                        /* entries                        */
} Pws_st_tables_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564158 754 ■

9 C PHIGS PLUS macro definitions

9.1 Function identifiers

The error functions require a unique mapping of the PHIGS PLUS functions to a set of numbers. The names for these function identifiers are the same as the bound PHIGS PLUS function names except that the sentinel character has been replaced by `Pfn_`.

Function Name Macro	Function Number
`#define Pfn_polyline_set3_colr`	(183)
`#define Pfn_fill_area_set3_data`	(184)
`#define Pfn_fill_area_set_data`	(185)
`#define Pfn_cell_array3_plus`	(186)
`#define Pfn_set_of_fill_area_sets3_data`	(187)
`#define Pfn_set_of_fill_area_sets_data`	(188)
`#define Pfn_tri_set3_data`	(189)
`#define Pfn_tri_set_data`	(190)
`#define Pfn_tri_strip3_data`	(191)
`#define Pfn_tri_strip_data`	(192)
`#define Pfn_quad_mesh3_data`	(193)
`#define Pfn_quad_mesh_data`	(194)
`#define Pfn_non_uniform_b_spline_curve3`	(195)
`#define Pfn_non_uniform_b_spline_curve3_colr`	(196)
`#define Pfn_non_uniform_b_spline_surf3`	(197)
`#define Pfn_non_uniform_b_spline_surf3_data`	(198)
`#define Pfn_set_data_map_ind`	(199)
`#define Pfn_set_refl_ind`	(200)
`#define Pfn_set_back_int_ind`	(201)
`#define Pfn_set_back_data_map_ind`	(202)
`#define Pfn_set_back_refl_ind`	(203)
`#define Pfn_set_param_surf_ind`	(204)
`#define Pfn_set_line_colr`	(205)
`#define Pfn_set_line_shad_method`	(206)
`#define Pfn_set_marker_colr`	(207)
`#define Pfn_set_text_colr`	(208)
`#define Pfn_set_facet_disting_mode`	(209)
`#define Pfn_set_facet_cull_mode`	(210)
`#define Pfn_set_int_colr`	(211)
`#define Pfn_set_int_shad_method`	(212)
`#define Pfn_set_data_map_method`	(213)
`#define Pfn_set_refl_props`	(214)
`#define Pfn_set_refl_model`	(215)
`#define Pfn_set_back_int_style`	(216)
`#define Pfn_set_back_int_style_ind`	(217)
`#define Pfn_set_back_int_colr`	(218)
`#define Pfn_set_back_int_shad_method`	(219)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_set_back_data_map_method          (220)
#define Pfn_set_back_refl_props               (221)
#define Pfn_set_back_refl_model               (222)
#define Pfn_set_light_source_st               (223)
#define Pfn_set_edge_colr                     (224)
#define Pfn_set_curve_approx_crit             (225)
#define Pfn_set_surf_approx_crit              (226)
#define Pfn_set_param_surf_chars              (227)
#define Pfn_set_rend_colr_model               (228)
#define Pfn_set_depth_cue_ind                 (229)
#define Pfn_set_colr_map_ind                  (230)
#define Pfn_set_line_rep_plus                 (231)
#define Pfn_set_marker_rep_plus               (232)
#define Pfn_set_text_rep_plus                 (233)
#define Pfn_set_int_rep_plus                  (234)
#define Pfn_set_edge_rep_plus                 (235)
#define Pfn_set_data_map_rep                  (236)
#define Pfn_set_refl_rep                      (237)
#define Pfn_set_param_surf_rep                (238)
#define Pfn_set_pat_rep_plus                  (239)
#define Pfn_set_light_source_rep              (240)
#define Pfn_set_depth_cue_rep                 (241)
#define Pfn_set_colr_map_rep                  (242)
```

9.2 Error codes

```
/* Output Attribute Errors */
#define PE_BAD_DEPTH_CUE_MODE          (119) /* Ignoring function, the spec-
                                                ified depth cue mode is not
                                                available on the workstation
                                                */
#define PE_DEPTH_CUE_INDX_LT_1         (120) /* Ignoring function, the
                                                depth cue index is less
                                                than one */
#define PE_COLR_MAP_INDX_LT_1          (121) /* Ignoring function, the
                                                colour mapping index is
                                                less than one */
#define PE_BAD_LINE_SHAD_METHOD        (122) /* Ignoring function, the spec-
                                                ified polyline shading
                                                method is not available on
                                                the workstation */
#define PE_BAD_INT_SHAD_METHOD         (123) /* Ignoring function, the spec-
                                                ified interior shading
                                                method is not available on
                                                the workstation */
#define PE_BAD_REFL_MODEL              (124) /* Ignoring function, the spec-
                                                ified reflectance model is
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564160 302 ■


```
                                                   not available on the work-
                                                   station */
#define PE_TOTAL_COLR_RANGE_FIELDS_TOO_LARGE   (125)
                                                /* Ignoring function, the
                                                   total of the colour range
                                                   fields in all the table
                                                   entries is too large */
#define PE_BAD_COLR_MAP_METHOD          (126) /* Ignoring function, the spec-
                                                   ified colour mapping method
                                                   is not available on the
                                                   specified workstation */
#define PE_BAD_APPROX_CRIT_TYPE         (127) /* Ignoring function, the spec-
                                                   ified approximation crite-
                                                   ria type is not available
                                                   on the specified workstation
                                                   */
#define PE_BAD_PARAM_SURF_CHAR_TYPE     (128) /* Ignoring function, the spec-
                                                   ified parametric surface
                                                   characteristics type is not
                                                   available on the specified
                                                   workstation */
#define PE_LIGHT_SOURCE_INDX_LT_1       (129) /* Ignoring function, the
                                                   light source index is less
                                                   than one */
#define PE_INVALID_REF_PLANES           (130) /* Ignoring function, invalid
                                                   reference planes; DQMIN >
                                                   DQMAX */
#define PE_BAD_LIGHT_SOURCE_TYPE        (131) /* Ignoring function, the spec-
                                                   ified light source type is
                                                   not available on the work-
                                                   station */
#define PE_SPOTLIGHT_SPREAD_ANGLE_RANGE (132) /* Ignoring function, the spec-
                                                   ified spot light spread
                                                   angle is out of range */
#define PE_BAD_ACTIVATION_LIST          (133) /* Ignoring function, one of
                                                   the entries in the activa-
                                                   tion list or deactivation
                                                   list is less than one */
#define PE_GEN_COLR_NOT_INDIRECT        (134) /* Ignoring function, the
                                                   requested entry contains a
                                                   general colour specifica-
                                                   tion with colour type other
                                                   than INDIRECT */
#define PE_SAME_ACTIVATION_DEACTIVATION_ENTRY  (135)
                                                /* Ignoring function, the same
                                                   entry exists in both the
                                                   activation and the deactiva-
                                                   tion list */
#define PE_COLR_COMPONENT_RANGE         (136) /* Ignoring function, one of
                                                   the components of the
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                                                    colour specification is out
                                                    of range */
#define PE_BAD_DATA_MAP_METHOD              (137) /* Ignoring function, the spec-
                                                    ified data mapping method
                                                    is not available on the
                                                    specified workstation */
#define PE_BAD_DATA_RECORD_FIELD            (138) /* Ignoring function, one or
                                                    more of the fields within
                                                    the specified data record
                                                    is in error */
#define PE_BAD_REFL_PROP_TYPE               (139) /* Ignoring function, the spec-
                                                    ified reflectance property
                                                    type is not available on
                                                    the specified workstation */
#define PE_INVALID_DEPTH_CUE_SCALE          (140) /* Ignoring function, one of
                                                    the depth cue scale factors
                                                    is not in the required range
                                                    */
/* Output Primitive Errors */
#define PE_ORDER_LT_1                       (500) /* Ignoring function, the
                                                    order of a spline is less
                                                    than one */
#define PE_INSUFF_CTRL_POINTS_FOR_ORDER     (501) /* Ignoring function, a spline
                                                    does not contain enough con-
                                                    trol points for its speci-
                                                    fied order */
#define PE_ORDER_INCONSISTENT_KNOTS_CTRL_POINTS   (502)
                                                  /* Ignoring function, the
                                                    order of a spline is incon-
                                                    sistent with number of
                                                    knots and control points */
#define PE_KNOT_SEQUENCE_NOT_NONDECREASING    (503)
                                                  /* Ignoring function, the knot
                                                    sequence for a spline is
                                                    not non-decreasing */
#define PE_VERTEX_INDX_OUT_OF_RANGE         (504) /* Ignoring function, one or
                                                    more of the vertex indices
                                                    is out of range */
#define PE_PARAM_RANGE_INCONSISTENT_KNOTS     (506)
                                                  /* Ignoring function, the
                                                    parameter range for a
                                                    spline is inconsistent with
                                                    its knots */
#define PE_W_VALUE_LE_0                     (507) /* Ignoring function, the
                                                    fourth coordinate of a
                                                    rational control point is
                                                    less than or equal to zero */
#define PE_TRIM_CURVE_ORDER_LT_2            (508) /* Ignoring function, a trim-
                                                    ming curve's order is less
                                                    than two */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_INCONSISTENT_EDGE_FLAG_SPEC   (513) /* Ignoring function, the num-
                                                  ber of edge visibility
                                                  flags is inconsistent with
                                                  the number of edges in the
                                                  primitive */
#define PE_DATA_LIST_LENGTH_NOT_SAME     (514) /* Ignoring function, the data
                                                  lists do not all contain
                                                  the same number of entries */
#define PE_INCONSISTENT_FACET_SPEC       (516) /* Ignoring function, the
                                                  facet data is inconsistent
                                                  with the number of facets
                                                  in the primitive */
#define PE_PARAM_RANGE_COLR_SPLINE       (517) /* Ignoring function, the
                                                  parameter range of the
                                                  colour spline does not
                                                  include the effective param-
                                                  eter range of the geometry
                                                  spline */
#define PE_PARAM_RANGE_DATA_SPLINE       (518) /* Ignoring function, the
                                                  parameter range of the data
                                                  spline does not include the
                                                  effective parameter range
                                                  of the geometry spline */
#define PE_BAD_VERTEX_DATA_TYPE          (2207)/* Ignoring function, two or
                                                  more vertices do not con-
                                                  tain the same type of data*/
```

9.3 Miscellaneous

9.3.1 Colour mapping methods

```
#define PCOLR_MAP_METHOD_TRUE            (1)
#define PCOLR_MAP_METHOD_PSEUDO          (2)
#define PCOLR_MAP_METHOD_PSEUDO_N        (3)
```

9.3.2 Colour types

```
#define PCOLR_INDIRECT                   (0)
#define PCOLR_RGB                        (1)
#define PCOLR_CIELUV                     (2)
#define PCOLR_HSV                        (3)
#define PCOLR_HLS                        (4)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9.3.3 Curve approximation criteria types

```
#define PCURVE_APPROX_WS_DEP                    (1)
#define PCURVE_APPROX_CONST_PARAM_SUBD          (2)
#define PCURVE_APPROX_CHORDAL_SIZE_WC           (3)
#define PCURVE_APPROX_CHORDAL_SIZE_NPC          (4)
#define PCURVE_APPROX_CHORDAL_SIZE_DC           (5)
#define PCURVE_APPROX_CHORDAL_DEV_WC            (6)
#define PCURVE_APPROX_CHORDAL_DEV_NPC           (7)
#define PCURVE_APPROX_CHORDAL_DEV_DC            (8)
#define PCURVE_APPROX_REL_WC                    (9)
#define PCURVE_APPROX_REL_NPC                   (10)
#define PCURVE_APPROX_REL_DC                    (11)
```

9.3.4 Data mapping methods

```
#define PDATA_MAP_METHOD_COLR                   (1)
#define PDATA_MAP_METHOD_SINGLE_UNIFORM         (2)
#define PDATA_MAP_METHOD_SINGLE_NON_UNIFORM     (3)
#define PDATA_MAP_METHOD_BI_UNIFORM             (4)
#define PDATA_MAP_METHOD_BI_NON_UNIFORM         (5)
```

9.3.5 Interior shading method

```
#define PINT_SHAD_METHOD_NONE                   (1)
#define PINT_SHAD_METHOD_COLR                   (2)
#define PINT_SHAD_METHOD_DATA                   (3)
#define PINT_SHAD_METHOD_DATA_DOT               (4)
#define PINT_SHAD_METHOD_DATA_NORM              (5)
```

9.3.6 Light source types

```
#define PLIGHT_SOURCE_AMB                       (1)
#define PLIGHT_SOURCE_DIR                       (2)
#define PLIGHT_SOURCE_POS                       (3)
#define PLIGHT_SOURCE_SPOT                      (4)
```

9.3.7 Parametric surface characteristic types

```
#define PPARAM_SURF_CHARS_NONE                  (1)
#define PPARAM_SURF_CHARS_WS_DEP                (2)
#define PPARAM_SURF_CHARS_ISOPARAM              (3)
#define PPARAM_SURF_CHARS_LEVEL_MC              (4)
#define PPARAM_SURF_CHARS_LEVEL_WC              (5)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564164 T58 ■

9.3.8 Polyline shading method

```
#define PLINE_SHAD_METHOD_NONE          (1)
#define PLINE_SHAD_METHOD_COLR          (2)
```

9.3.9 Reflectance models

```
#define PREFL_MODEL_NO_REFL             (1)
#define PREFL_MODEL_AMB_REFL            (2)
#define PREFL_MODEL_AMB_DIFF_REFL       (3)
#define PREFL_MODEL_AMB_DIFF_SPEC_REFL  (4)
```

9.3.10 Reflectance properties

```
#define PREFL_PROPS_SIMPLE_REFL         (1)
```

9.3.11 Rendering Colour Model

```
#define PREND_COLR_MODEL_WS_DEP         (0)
#define PREND_COLR_MODEL_RGB            (1)
#define PREND_COLR_MODEL_CIELUV         (2)
#define PREND_COLR_MODEL_HSV            (3)
#define PREND_COLR_MODEL_HLS            (4)
```

9.3.12 Surface approximation criteria types

```
#define PSURF_APPROX_WS_DEP             (1)
#define PSURF_APPROX_CONST_PARAM_SUBD   (2)
#define PSURF_APPROX_CHORDAL_SIZE_WC    (3)
#define PSURF_APPROX_CHORDAL_SIZE_NPC   (4)
#define PSURF_APPROX_CHORDAL_SIZE_DC    (5)
#define PSURF_APPROX_PLANAR_DEV_WC      (6)
#define PSURF_APPROX_PLANAR_DEV_NPC     (7)
#define PSURF_APPROX_PLANAR_DEV_DC      (8)
#define PSURF_APPROX_REL_WC             (9)
#define PSURF_APPROX_REL_NPC            (10)
#define PSURF_APPROX_REL_DC             (11)
```

9.3.13 Element enumeration

```
#define PFIRST_PHIGS_PLUS_ELEM          (PELEM_POLYLINE_SET3_COLR)
#define PLAST_PHIGS_PLUS_ELEM           (PELEM_COLR_MAP_IND)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564165 994 ■

10 C PHIGS PLUS functions

10.1 Output primitive functions

POLYLINE SET 3 WITH COLOUR

```
void ppolyline_set3_colr (

  Pint                    colr_type,      /* colour type                  */
  const Pvertex3_set      *vertex_data    /* list of vertex lists         */

);
```

NOTE — Only the vertex points and colour portions of the data pointed to by the `vertex_data` parameter are used. All other data is ignored.

FILL AREA SET 3 WITH DATA

```
void pfill_area_set3_data (

  Pint                    colr_type,      /* colour type                  */
  const Pfacet            *facet_data,    /* facet data                   */
  const Pedge_flag_set    *edge_flags,    /* list of edge flag lists      */
  const Pvertex3_set      *vertex_data    /* list of vertex lists         */

);
```

NOTE — A `NULL` value for the `facet_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

FILL AREA SET WITH DATA

```
void pfill_area_set_data (

  Pint                    colr_type,      /* colour type                  */
  const Pfacet            *facet_data,    /* facet data                   */
  const Pedge_flag_set    *edge_flags,    /* list of edge flag lists      */
  const Pvertex_set       *vertex_data    /* list of vertex lists         */

);
```

NOTE — A NULL value for the facet_data or edge_flags parameters indicates the corresponding optional data is not supplied.

CELL ARRAY 3 PLUS

```
void pcell_array3_plus (

  const Pparal            *paral,         /* parallelogram P, Q, R        */
  const Ppat_rep_plus     *pat            /* pattern colour array         */

);
```

SET OF FILL AREA SETS 3 WITH DATA

```
void pset_of_fill_area_sets3_data (

  Pint                        colr_type,    /* colour type                */
  const Pset_of_fill_area_sets_data  *set_data,/*fill area set data        */
  const Pedge_flag_set_list   *edge_flags,  /* list of edge flag sets       */
  const Pvertex3_list         *vertex_data, /* list of vertices             */
  const Pcontour_set_list     *contours     /* list of contour sets         */

);
```

NOTE — A NULL value for the set_data or edge_flags parameters indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET OF FILL AREA SETS WITH DATA

```
void pset_of_fill_area_sets_data (

  Pint                        colr_type,    /* colour type                    */
  const Pset_of_fill_area_sets_data  *set_data,/*fill area set data           */
  const Pedge_flag_set_list   *edge_flags,  /* list of edge flag sets         */
  const Pvertex_list          *vertex_data, /* list of vertices               */
  const Pcontour_set_list     *contours     /* list of contour sets           */

);
```

NOTE — A NULL value for the `set_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

TRIANGLE SET 3 WITH DATA

```
void ptri_set3_data (

  Pint                      colr_type,      /* colour type                    */
  const Ptri_set_data       *set_data,      /* triangle set data              */
  const Pedge_flag_triplet_list *edge_flags,/*list of triangle edge flag      */
                                            /* triplets                       */
  const Pvertex3_list       *vertex_data,   /* list of vertices               */
  const Pvertex_ind_triplet_list *triplets /* list of triangle vertex index   */
                                            /* triplets                       */

);
```

NOTE — A NULL value for the `set_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

TRIANGLE SET WITH DATA

```
void ptri_set_data (

  Pint                        colr_type,      /* colour type                    */
  const Ptri_set_data         *set_data,      /* triangle set data              */
  const Pedge_flag_triplet_list *edge_flags,/*list of triangle edge flag      */
                                              /* triplets                       */
  const Pvertex_list          *vertex_data,   /* list of vertices               */
  const Pvertex_ind_triplet_list *vertex_inds/*list of triangle vertex index */
                                              /* triplets                       */

);
```

NOTE — A NULL value for the `set_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

TRIANGLE STRIP 3 WITH DATA

```
void ptri_strip3_data (

  Pint                      colr_type,        /* colour type                    */
  const Pfacet_list         *facet_data,      /* list of facet data             */
  const Pedge_flag_list     *edge_flags,      /* list of edge flags             */
  const Pvertex3_list       *vertex_data      /* list of vertex data            */

);
```

NOTE — A NULL value for the `facet_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

TRIANGLE STRIP WITH DATA

```
void ptri_strip_data (

  Pint                      colr_type,        /* colour type                    */
  const Pfacet_list         *facet_data,      /* list of facet data             */
  const Pedge_flag_list     *edge_flags,      /* list of edge flags             */
  const Pvertex_list        *vertex_data      /* list of vertex data            */

);
```

NOTE — A NULL value for the `facet_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

QUADRILATERAL MESH 3 WITH DATA

```
void pquad_mesh3_data (

  Pint                    colr_type,      /* colour type                */
  const Pfacet_array      *facet_data,    /* array of facet data        */
  const Pedge_flag_array  *edge_flags,    /* array of edge flags        */
  const Pvertex3_array    *vertex_data    /* array of vertex data       */

);
```

NOTE — A NULL value for the `facet_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

QUADRILATERAL MESH WITH DATA

```
void pquad_mesh_data (

  Pint                    colr_type,      /* colour type                */
  const Pfacet_array      *facet_data,    /* array of facet data        */
  const Pedge_flag_array  *edge_flags,    /* array of edge flags        */
  const Pvertex_array     *vertex_data    /* array of vertex data       */

);
```

NOTE — A NULL value for the `facet_data` or `edge_flags` parameters indicates the corresponding optional data is not supplied.

NON-UNIFORM B-SPLINE CURVE 3

```
void pnon_uniform_b_spline_curve3 (

  const Pcurve_geom_spline  *geom_spline   /* curve geometry spline      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR

```
void pnon_uniform_b_spline_curve3_colr (

  const Pcurve_geom_spline  *geom_spline   /* curve geometry spline          */
  const Pcurve_colr_spline  *colr_spline   /* curve colour spline            */

);
```

NOTE — A NULL value for the `colr_spline` parameter indicates the corresponding optional data is not supplied.

NON-UNIFORM B-SPLINE SURFACE 3

```
void pnon_uniform_b_spline_surf3 (

  const Psurf_geom_spline      *geom_spline,/*surface geometry spline        */
  const Ptrim_curve_loop_list  *trim_loops /* list of trimming curve loops   */

);
```

NOTE — A NULL value for the `trim_loops` parameter indicates the corresponding optional data is not supplied.

NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA

```
void pnon_uniform_b_spline_surf3_data (

  const Psurf_geom_spline       *geom_spline,/*surface geometry spline      */
  const Ptrim_curve_loop_list   *trim_loops,/*list of trimming curve loops  */
  const Psurf_colr_spline       *colr_spline,/*colour spline                */
  const Psurf_data_spline_list  *data_splines/*list of data splines         */

);
```

NOTE — A NULL value for the `trim_loops`, `colr_spline`, and `data_splines` parameters indicates the corresponding optional data is not supplied.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.2 Attribute specification functions

10.2.1 Bundled attribute selection

SET DATA MAPPING INDEX

```
void pset_data_map_ind (
  Pint                  index                 /* data mapping index              */
);
```

SET REFLECTANCE INDEX

```
void pset_refl_ind (
  Pint                  index                 /* reflectance index               */
);
```

SET BACK INTERIOR INDEX

```
void pset_back_int_ind (
  Pint                  index                 /* back interior index             */
);
```

SET BACK DATA MAPPING INDEX

```
void pset_back_data_map_ind (
  Pint                  index                 /* back data mapping index         */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET BACK REFLECTANCE INDEX

```
void pset_back_refl_ind (
  Pint                index                /* back reflectance index        */
);
```

SET PARAMETRIC SURFACE INDEX

```
void pset_param_surf_ind (
  Pint                index                /* parametric surface index      */
);
```

10.2.2 Individual attribute selection

SET POLYLINE COLOUR

```
void pset_line_colr (
  const Pgcolr        *colr                /* polyline colour               */
);
```

SET POLYLINE SHADING METHOD

```
void pset_line_shad_method (
  Pint                method               /* polyline shading method       */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET POLYMARKER COLOUR

```
void pset_marker_colr (
  const Pgcolr        *colr                /* polymarker colour              */
);
```

SET TEXT COLOUR

```
void pset_text_colr (
  const Pgcolr        *colr                /* text colour                    */
);
```

SET FACET DISTINGUISHING MODE

```
void pset_facet_disting_mode (
  Pfacet_disting_mode      mode            /* distinguishing mode            */
);
```

SET FACET CULLING MODE

```
void pset_facet_cull_mode (
  Pfacet_cull_mode    mode                 /* facet culling mode             */
);
```

SET INTERIOR COLOUR

```
void pset_int_colr (
  const Pgcolr        *colr                /* interior colour                */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

SET INTERIOR SHADING METHOD

```
void pset_int_shad_method (
  Pint                method                /* interior shading method        */
);
```

SET DATA MAPPING METHOD

```
void pset_data_map_method (
  const Pdata_map_rep     *method           /* data mapping method            */
);
```

SET REFLECTANCE PROPERTIES

```
void pset_refl_props (
  const Prefl_prop    *refl_prop            /* reflectance property data      */
);
```

SET REFLECTANCE MODEL

```
void pset_refl_model (
  Pint                model                 /* reflectance model              */
);
```

SET BACK INTERIOR STYLE

```
void pset_back_int_style (
  Pint_style          style                 /* back interior style            */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564175 833 ■

SET BACK INTERIOR STYLE INDEX

```
void pset_back_int_style_ind (
  Pint                  index                  /* back interior style index      */
);
```

SET BACK INTERIOR COLOUR

```
void pset_back_int_colr (
  const Pgcolr          *colr                  /* back interior colour           */
);
```

SET BACK INTERIOR SHADING METHOD

```
void pset_back_int_shad_method (
  Pint                  method                 /* interior shading method        */
);
```

SET BACK DATA MAPPING METHOD

```
void pset_back_data_map_method (
  const Pdata_map_rep     *method              /* data mapping method            */
);
```

SET BACK REFLECTANCE PROPERTIES

```
void pset_back_refl_props (
  const Prefl_prop        *refl_data           /* reflectance property data      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564176 77T ■

SET BACK REFLECTANCE MODEL

```
void pset_back_refl_model (
  Pint                model                /* reflectance model              */
);
```

SET LIGHT SOURCE STATE

```
void pset_light_source_st (
  const Pint_list     *act_list,           /* activation list                */
  const Pint_list     *deact_list          /* deactivation list              */
);
```

SET EDGE COLOUR

```
void pset_edge_colr (
  const Pgcolr        *colr                /* edge colour                    */
);
```

SET CURVE APPROXIMATION CRITERIA

```
void pset_curve_approx_crit (
  const Pcurve_approx_crit   *data         /* curve approximation criteria   */
                                           /* data record                    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET SURFACE APPROXIMATION CRITERIA

```
void pset_surf_approx_crit (
  const Psurf_approx_crit    *data          /* surface approximation criteria */
                                            /* data record                    */
);
```

SET PARAMETRIC SURFACE CHARACTERISTICS

```
void pset_param_surf_chars (
  const Pparam_surf_chars    *data          /* parametric surface             */
                                            /* characteristics data record    */
);
```

SET RENDERING COLOUR MODEL

```
void pset_rend_colr_model (
  Pint                 model                /* rendering colour model         */
);
```

SET DEPTH CUE INDEX

```
void pset_depth_cue_ind (
  Pint                 index                /* depth cue index                */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET COLOUR MAPPING INDEX

```
void pset_colr_map_ind (
  Pint                index                /* colour mapping index          */
);
```

10.2.3 Workstation attribute table definition

SET POLYLINE REPRESENTATION PLUS

```
void pset_line_rep_plus (
  Pint                        ws_id,       /* workstation identifier        */
  Pint                        index,       /* polyline bundle index         */
  const Pline_bundle_plus     *rep         /* polyline representation       */
);
```

SET POLYMARKER REPRESENTATION PLUS

```
void pset_marker_rep_plus (
  Pint                        ws_id,       /* workstation identifier        */
  Pint                        index,       /* polymarker bundle index       */
  const Pmarker_bundle_plus   *rep         /* polymarker representation     */
);
```

SET TEXT REPRESENTATION PLUS

```
void pset_text_rep_plus (
  Pint                        ws_id,       /* workstation identifier        */
  Pint                        index,       /* text bundle index             */
  const Ptext_bundle_plus     *rep         /* text representation           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564179 489 ■

SET INTERIOR REPRESENTATION PLUS

```
void pset_int_rep_plus (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* interior bundle index           */
  const Pint_bundle_plus      *rep          /* interior representation         */

);
```

SET EDGE REPRESENTATION PLUS

```
void pset_edge_rep_plus (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* edge bundle index               */
  const Pedge_bundle_plus     *rep          /* edge representation             */

);
```

SET DATA MAPPING REPRESENTATION

```
void pset_data_map_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* data mapping index              */
  const Pdata_map_rep         *rep          /* data mapping representation     */

);
```

SET REFLECTANCE REPRESENTATION

```
void pset_refl_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* reflectance index               */
  const Prefl_rep             *rep          /* reflectance representation      */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET PARAMETRIC SURFACE REPRESENTATION

```
void pset_param_surf_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* parametric surface rep. index   */
  const Pparam_surf_rep       *rep          /* parametric surface              */
                                            /* representation                  */

);
```

SET PATTERN REPRESENTATION PLUS

```
void pset_pat_rep_plus (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* pattern index                   */
  const Ppat_rep_plus         *rep          /* pattern representation plus     */

);
```

SET LIGHT SOURCE REPRESENTATION

```
void pset_light_source_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* light source index              */
  const Plight_source_rep     *rep          /* light source representation     */

);
```

SET DEPTH CUE REPRESENTATION

```
void pset_depth_cue_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* depth cue index                 */
  const Pdepth_cue_rep        *rep          /* depth cue representation        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET COLOUR MAPPING REPRESENTATION

```
void pset_colr_map_rep (

  Pint                        ws_id,        /* workstation identifier          */
  Pint                        index,        /* colour mapping index            */
  const Pcolr_map_rep         *rep          /* colour mapping representation   */

);
```

10.3 Inquiry functions

10.3.1 Inquiry functions for the workstation state list

INQUIRE POLYLINE REPRESENTATION PLUS

```
void pinq_line_rep_plus (

  Pint                ws_id,              /* workstation identifier          */
  Pint                index,              /* polyline index                  */
  Pinq_type           type,               /* type of returned values         */
  Pint                *err_ind,           /* OUT error indicator             */
  Pline_bundle_plus   *rep                /* OUT polyline representation     */

);
```

INQUIRE POLYMARKER REPRESENTATION PLUS

```
void pinq_marker_rep_plus (

  Pint                    ws_id,          /* workstation identifier          */
  Pint                    index,          /* polymarker index                */
  Pinq_type               type,           /* type of returned values         */
  Pint                    *err_ind,       /* OUT error indicator             */
  Pmarker_bundle_plus     *rep            /* OUT polymarker representation   */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TEXT REPRESENTATION PLUS

```
void pinq_text_rep_plus (

  Pint                ws_id,              /* workstation identifier        */
  Pint                index,              /* text index                    */
  Pinq_type           type,               /* type of returned values       */
  Pint                *err_ind,           /* OUT error indicator           */
  Ptext_bundle_plus   *rep                /* OUT text representation       */

);
```

INQUIRE INTERIOR REPRESENTATION PLUS

```
void pinq_int_rep_plus (

  Pint                ws_id,              /* workstation identifier        */
  Pint                index,              /* interior index                */
  Pinq_type           type,               /* type of returned values       */
  Pint                *err_ind,           /* OUT error indicator           */
  Pint_bundle_plus    *rep                /* OUT interior representation   */

);
```

INQUIRE EDGE REPRESENTATION PLUS

```
void pinq_edge_rep_plus (

  Pint                ws_id,              /* workstation identifier        */
  Pint                index,              /* edge index                    */
  Pinq_type           type,               /* type of returned values       */
  Pint                *err_ind,           /* OUT error indicator           */
  Pedge_bundle_plus   *rep                /* OUT edge representation       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LIST OF DATA MAPPING INDICES

```
void pinq_list_data_map_inds (

  Pint              ws_id,               /* workstation identifier         */
  Pint              num_elems_appl_list, /* # elems in appl list           */
  Pint              start_ind,           /* starting index                 */
  Pint              *err_ind,            /* OUT error indicator            */
  Pint_list         *indices,            /* OUT list of defined data       */
                                         /* mapping indices                */
  Pint              *num_elems_impl_list /* OUT # elems in impl list       */

);
```

INQUIRE DATA MAPPING REPRESENTATION

```
void pinq_data_map_rep (

  Pint              ws_id,               /* workstation identifier         */
  Pint              index,               /* data mapping index             */
  Pinq_type         type,                /* type of returned values        */
  Pstore            store,               /* handle to Store object         */
  Pint              *err_ind,            /* OUT error indicator            */
  Pdata_map_rep     **rep                /* OUT data mapping representation*/

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE LIST OF REFLECTANCE INDICES

```
void pinq_list_refl_inds (

  Pint              ws_id,               /* workstation identifier         */
  Pint              num_elems_appl_list, /* # elems in appl list           */
  Pint              start_ind,           /* starting index                 */
  Pint              *err_ind,            /* OUT error indicator            */
  Pint_list         *indices,            /* OUT list of defined reflectance*/
                                         /* indices                        */
  Pint              *num_elems_impl_list /* OUT # elems in impl list       */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE REFLECTANCE REPRESENTATION

```
void pinq_refl_rep (

  Pint              ws_id,              /* workstation identifier         */
  Pint              index,              /* reflectance index              */
  Pinq_type         type,               /* type of returned values        */
  Pstore            store,              /* handle to Store object         */
  Pint              *err_ind,           /* OUT error indicator            */
  Prefl_rep         **rep               /* OUT reflectance representation */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE LIST OF PARAMETRIC SURFACE INDICES

```
void pinq_list_param_surf_inds (

  Pint              ws_id,              /* workstation identifier         */
  Pint              num_elems_appl_list, /* # elems in appl list          */
  Pint              start_ind,          /* starting index                 */
  Pint              *err_ind,           /* OUT error indicator            */
  Pint_list         *indices,           /* OUT list of defined parametric */
                                        /* surface indices                */
  Pint              *num_elems_impl_list /* OUT # elems in impl list      */

);
```

INQUIRE PARAMETRIC SURFACE REPRESENTATION

```
void pinq_param_surf_rep (

  Pint              ws_id,              /* workstation identifier         */
  Pint              index,              /* parametric surface bundle index*/
  Pinq_type         type,               /* type of returned values        */
  Pstore            store,              /* handle to Store object         */
  Pint              *err_ind,           /* OUT error indicator            */
  Pparam_surf_rep   **rep               /* OUT parametric surface         */
                                        /* representation                 */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564185 782 ■

INQUIRE PATTERN REPRESENTATION PLUS

```
void pinq_pat_rep_plus (

  Pint              ws_id,              /* workstation identifier          */
  Pint              index,              /* pattern index                   */
  Pinq_type         type,               /* type of returned values         */
  Pstore            store,              /* handle to Store object          */
  Pint              *err_ind,           /* OUT error indicator             */
  Ppat_rep_plus     **rep               /* OUT pattern representation plus*/

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE LIST OF LIGHT SOURCE INDICES

```
void pinq_list_light_source_inds (

  Pint              ws_id,              /* workstation identifier          */
  Pint              num_elems_appl_list, /* # elems in appl list           */
  Pint              start_ind,          /* starting index                  */
  Pint              *err_ind,           /* OUT error indicator             */
  Pint_list         *indices,           /* OUT list of defined light       */
                                        /* source indices                  */
  Pint              *num_elems_impl_list /* OUT # elems in impl list       */

);
```

INQUIRE LIGHT SOURCE REPRESENTATION

```
void pinq_light_source_rep (

  Pint              ws_id,              /* workstation identifier          */
  Pint              index,              /* light source index              */
  Pinq_type         type,               /* type of returned values         */
  Pint              *err_ind,           /* OUT error indicator             */
  Plight_source_rep  *rep               /* OUT light source representation*/

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE LIST OF DEPTH CUE INDICES

```
void pinq_list_depth_cue_inds (

  Pint              ws_id,               /* workstation identifier          */
  Pint              num_elems_appl_list, /* # elems in appl list            */
  Pint              start_ind,           /* starting index                  */
  Pint              *err_ind,            /* OUT error indicator             */
  Pint_list         *indices,            /* OUT list of defined depth cue   */
                                         /* indices                         */
  Pint              *num_elems_impl_list /* OUT # elems in impl list        */

);
```

INQUIRE DEPTH CUE REPRESENTATION

```
void pinq_depth_cue_rep (

  Pint              ws_id,               /* workstation identifier          */
  Pint              index,               /* depth cue index                 */
  Pinq_type         type,                /* type of returned values         */
  Pint              *err_ind,            /* OUT error indicator             */
  Pdepth_cue_rep    *rep                 /* OUT depth cue representation    */

);
```

INQUIRE COLOUR MAPPING STATE

```
void pinq_colr_map_st (

  Pint              ws_id,               /* workstation identifier          */
  Pint              method,              /* colour mapping method           */
  Pstore            store,               /* handle to Store object          */
  Pint              *err_ind,            /* OUT error indicator             */
  Pcolr_map_st      **data               /* OUT data record                 */

);
```

NOTE — The memory referenced by `*data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564187 555 ■

INQUIRE LIST OF COLOUR MAPPING INDICES

```
void pinq_list_colr_map_inds (

  Pint              ws_id,               /* workstation identifier        */
  Pint              num_elems_appl_list, /* # elems in appl list          */
  Pint              start_ind,           /* starting index                */
  Pint              *err_ind,            /* OUT error indicator           */
  Pint_list         *indices,            /* OUT list of defined colour    */
                                         /* mapping indices               */
  Pint              *num_elems_impl_list /* OUT # elems in impl list      */

);
```

INQUIRE COLOUR MAPPING REPRESENTATION

```
void pinq_colr_map_rep (

  Pint              ws_id,               /* workstation identifier        */
  Pint              index,               /* colour mapping index          */
  Pinq_type         type,                /* type of returned values       */
  Pstore            store,               /* handle to Store object        */
  Pint              *err_ind,            /* OUT error indicator           */
  Pcolr_map_rep     **rep                /* OUT colour mapping            */
                                         /* representation                */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

10.3.2 Inquiry functions for the workstation description table

---

INQUIRE DIRECT COLOUR MODEL FACILITIES

```
void pinq_direct_colr_model_facs (

  Pint           ws_type,                 /* workstation type                */
  Pint           num_elems_appl_list,     /* # elems in appl list            */
  Pint           start_ind,               /* starting index                  */
  Pint           *err_ind,                /* OUT error indicator             */
  Pint_list      *models,                 /* OUT list of directly            */
                                          /* specifiable colour models       */
  Pint           *num_elems_impl_list     /* OUT # elems in impl list        */

);
```

INQUIRE RENDERING COLOUR MODEL FACILITIES

```
void pinq_rend_colr_model_facs (

  Pint           ws_type,                 /* workstation type                */
  Pint           num_elems_appl_list,     /* # elems in appl list            */
  Pint           start_ind,               /* starting index                  */
  Pint           *err_ind,                /* OUT error indicator             */
  Pint_list      *models,                 /* OUT list of available rendering*/
                                          /* colour models                   */
  Pint           *num_elems_impl_list     /* OUT # elems in impl list        */

);
```

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS

```
void pinq_dyns_ws_attrs_plus (

  Pint                   ws_type,         /* workstation type                */
  Pint                   *err_ind,        /* OUT error indicator             */
  Pdyn_ws_attrs_plus     *attr            /* OUT dynamics of workstation     */
                                          /* attributes plus                 */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE POLYLINE FACILITIES PLUS

```
void pinq_line_facs_plus (

  Pint                ws_type,              /* workstation type              */
  Pstore              store,                /* handle to Store object        */
  Pint                *err_ind,             /* OUT error indicator           */
  Pline_facs_plus     **facs                /* OUT list of polyline          */
                                            /* facilities plus               */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS

```
void pinq_pred_line_rep_plus (

  Pint                    ws_type,          /* workstation type              */
  Pint                    index,            /* predefined polyline index     */
  Pint                    *err_ind,         /* OUT error indicator           */
  Pline_bundle_plus       *rep              /* OUT predefined polyline       */
                                            /* representation                */

);
```

INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS

```
void pinq_pred_marker_rep_plus (

  Pint                    ws_type,          /* workstation type              */
  Pint                    index,            /* predefined polymarker index   */
  Pint                    *err_ind,         /* OUT error indicator           */
  Pmarker_bundle_plus     *rep              /* OUT predefined polymarker     */
                                            /* representation                */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED TEXT REPRESENTATION PLUS

```
void pinq_pred_text_rep_plus (

  Pint                ws_type,          /* workstation type                */
  Pint                index,            /* predefined text index           */
  Pint                *err_ind,         /* OUT error indicator             */
  Ptext_bundle_plus   *rep              /* OUT predefined text             */
                                        /* representation                  */

);
```

INQUIRE INTERIOR FACILITIES PLUS

```
void pinq_int_facs_plus (

  Pint                ws_type,          /* workstation type                */
  Pstore              store,            /* handle to Store object          */
  Pint                *err_ind,         /* OUT error indicator             */
  Pint_facs_plus      **facs            /* OUT list of interior            */
                                        /* facilities plus                 */

);
```

NOTE — The memory referenced by `*facs` is managed by `store`.

INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS

```
void pinq_pred_int_rep_plus (

  Pint                ws_type,          /* workstation type                */
  Pint                index,            /* predefined interior index       */
  Pint                *err_ind,         /* OUT error indicator             */
  Pint_bundle_plus    *rep              /* OUT predefined interior         */
                                        /* representation                  */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564191 T86 ■

INQUIRE PREDEFINED EDGE REPRESENTATION PLUS

```
void pinq_pred_edge_rep_plus (

  Pint                ws_type,            /* workstation type                */
  Pint                index,              /* predefined edge index           */
  Pint                *err_ind,           /* OUT error indicator             */
  Pedge_bundle_plus   *rep                /* OUT predefined edge             */
                                          /* representation                  */

);
```

INQUIRE DATA MAPPING FACILITIES

```
void pinq_data_map_facs (

  Pint           ws_type,                 /* workstation type                */
  Pint           num_elems_appl_list,     /* # elems in appl list            */
  Pint           start_ind,               /* starting index                  */
  Pint           *err_ind,                /* OUT error indicator             */
  Pint_list      *methods,                /* OUT list of available data      */
                                          /* mapping methods                 */
  Pint           *num_inds,               /* OUT number of predefined data   */
                                          /* mapping indices                 */
  Pint           *num_elems_impl_list     /* OUT # elems in impl list        */

);
```

INQUIRE PREDEFINED DATA MAPPING REPRESENTATION

```
void pinq_pred_data_map_rep (

  Pint                ws_type,            /* workstation type                */
  Pint                index,              /* predefined data mapping index   */
  Pstore              store,              /* handle to Store object          */
  Pint                *err_ind,           /* OUT error indicator             */
  Pdata_map_rep       **rep               /* OUT data mapping representation*/

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE REFLECTANCE FACILITIES

```
void pinq_refl_facs (

  Pint          ws_type,              /* workstation type              */
  Pstore        store,                /* handle to Store object        */
  Pint          *err_ind,             /* OUT error indicator           */
  Pint_list     **models,             /* OUT list of available         */
                                      /* reflectance models            */
  Pint_list     **refl_props,         /* OUT list of available         */
                                      /* reflectance properties        */
  Pint          *refl_indices         /* OUT number of predefined      */
                                      /* reflectance indices           */

);
```

NOTE — The memory referenced by `*models` and `*refl_props` is managed by `store`.

INQUIRE PREDEFINED REFLECTANCE REPRESENTATION

```
void pinq_pred_refl_rep (

  Pint          ws_type,              /* workstation type              */
  Pint          index,                /* predefined reflectance index  */
  Pstore        store,                /* handle to Store object        */
  Pint          *err_ind,             /* OUT error indicator           */
  Prefl_rep     **rep                 /* OUT predefined reflectance    */
                                      /* representation                */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE CURVE AND SURFACE FACILITIES

The **INQUIRE CURVE AND SURFACE FACILITIES** is broken into four related functions: **INQUIRE CURVE FACILITIES, INQUIRE PARAMETRIC SURFACE FACILITIES, INQUIRE B-SPLINE SURFACE FACILITIES,** and **INQUIRE TRIMMING CURVE FACILITIES.**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE CURVE FACILITIES

```
void pinq_curve_facs (

  Pint          ws_type,                /* workstation type                */
  Pint          num_elems_appl_list,    /* # elems in appl list            */
  Pint          start_ind,              /* starting index                  */
  Pint          *err_ind,               /* OUT error indicator             */
  Pint          *max_order,             /* OUT maximum non-uniform         */
                                        /* b-spline                        */
                                        /* curve order supported           */
  Pint_list     *types,                 /* OUT list of available curve     */
                                        /* approximation criteria types    */
  Pint          *num_elems_impl_list    /* OUT # elems in impl list        */

);
```

INQUIRE PARAMETRIC SURFACE FACILITIES

```
void pinq_param_surf_facs (

  Pint          ws_type,                /* workstation type                */
  Pint          num_elems_appl_list,    /* # elems in appl list            */
  Pint          start_ind,              /* starting index                  */
  Pint          *err_ind,               /* OUT error indicator             */
  Pint_list     *types,                 /* OUT list of available           */
                                        /* parametric surface              */
                                        /* characteristic types            */
  Pint          *num_elems_impl_list,   /* OUT # elems in impl list        */
  Pint          *num_indices            /* OUT number of predefined        */
                                        /* parametric surface indices      */

);
```

INQUIRE B-SPLINE SURFACE FACILITIES

```
void pinq_b_spline_surf_facs (

  Pint          ws_type,                /* workstation type                */
  Pint          num_elems_appl_list,    /* # elems in appl list            */
  Pint          start_ind,              /* starting index                  */
  Pint          *err_ind,               /* OUT error indicator             */
  Pint          *max_order,             /* OUT maximum non-uniform         */
                                        /* b-spline surface order          */
                                        /* supported                       */
  Pint_list     *types,                 /* OUT list of available surface   */
                                        /* approximation criteria types    */
  Pint          *num_elems_impl_list    /* OUT # elems in impl list        */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE TRIMMING CURVE FACILITIES

```
void pinq_trim_curve_facs (

  Pint          ws_type,                  /* workstation type                */
  Pint          num_elems_appl_list,      /* # elems in appl list            */
  Pint          start_ind,                /* starting index                  */
  Pint          *err_ind,                 /* OUT error indicator             */
  Pint          *max_order,               /* OUT maximum trimming curve      */
                                          /* order supported                 */
  Pint_list     *types,                   /* OUT list of available trimming  */
                                          /* curve approximation criteria    */
                                          /* types                           */
  Pint          *num_elems_impl_list      /* OUT # elems in impl list        */

);
```

INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION

```
void pinq_pred_param_surf_rep (

  Pint              ws_type,              /* workstation type                */
  Pint              index,                /* predefined parametric surface   */
                                          /* index                           */
  Pstore            store,                /* handle to Store object          */
  Pint              *err_ind,             /* OUT error indicator             */
  Pparam_surf_rep   **rep                 /* OUT parametric surface          */
                                          /* representation                  */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS

```
void pinq_pred_pat_rep_plus (

  Pint              ws_type,              /* workstation type                */
  Pint              index,                /* predefined pattern index        */
  Pstore            store,                /* handle to Store object          */
  Pint              *err_ind,             /* OUT error indicator             */
  Ppat_rep_plus     **rep                 /* OUT pattern representation plus*/

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE LIGHT SOURCE FACILITIES

```
void pinq_light_source_facs (

  Pint           ws_type,                 /* workstation type                */
  Pint           num_elems_appl_list,     /* # elems in appl list            */
  Pint           start_ind,               /* starting index                  */
  Pint           *err_ind,                /* OUT error indicator             */
  Pint_list      *types,                  /* OUT list of available light     */
                                          /* source types                    */
  Pint           *num_elems_impl_list,    /* OUT # elems in impl list        */
  Pint           *max_lights,             /* OUT maximum number of           */
                                          /* simultaneously active           */
                                          /* non-ambient lights              */
  Pint           *num_indices             /* OUT number of predefined light  */
                                          /* source indices                  */

);
```

INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION

```
void pinq_pred_light_source_rep (

  Pint                    ws_type,        /* workstation type                */
  Pint                    index,          /* predefined light source index   */
  Pint                    *err_ind,       /* OUT error indicator             */
  Plight_source_rep       *rep            /* OUT light source representation*/

);
```

INQUIRE DEPTH CUE FACILITIES

```
void pinq_depth_cue_facs (

  Pint                  ws_type,          /* workstation type                */
  Pint                  num_elems_appl_list,/* # elems in appl list          */
  Pint                  start_ind,        /* starting index                  */
  Pint                  *err_ind,         /* OUT error indicator             */
  Pint                  *num_indices      /* OUT number of predefined depth  */
                                          /* cue indices                     */
  Pdepth_cue_mode_list  *modes,           /* OUT list of available depth     */
                                          /* cue modes                       */
  Pint                  *num_elems_impl_list/* OUT # elems impl list         */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE PREDEFINED DEPTH CUE REPRESENTATION

```
void pinq_pred_depth_cue_rep (

  Pint                ws_type,              /* workstation type                */
  Pint                index,                /* predefined depth cue index      */
  Pint                *err_ind,             /* OUT error indicator             */
  Pdepth_cue_rep      *rep                  /* OUT depth cue representation    */

);
```

INQUIRE COLOUR MAPPING FACILITIES

```
void pinq_colr_map_facs (

  Pint            ws_type,                  /* workstation type                */
  Pint            num_elems_appl_list,      /* # elems in appl list            */
  Pint            start_ind,                /* starting index                  */
  Pint            *err_ind,                 /* OUT error indicator             */
  Pint_list       *methods,                 /* OUT list of available colour    */
                                            /* mapping methods                 */
  Pint            *num_elems_impl_list      /* OUT # elems in impl list        */
  Pint            *num_indices              /* OUT number of predefined colour*/
                                            /* mapping indices                 */

);
```

INQUIRE COLOUR MAPPING METHOD FACILITIES

```
void pinq_colr_map_method_facs (

  Pint                    ws_type,          /* workstation type                */
  Pint                    method,           /* colour mapping methods          */
  Pstore                  store,            /* handle to Store object          */
  Pint                    *err_ind,         /* OUT error indicator             */
  Pcolr_map_method        **data            /* OUT data record                 */

);
```

NOTE — The memory referenced by `*data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564197 4T4 ■

INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION

```
void pinq_pred_colr_map_rep (

  Pint                ws_type,            /* workstation type                */
  Pint                index,              /* predefined colour mapping index*/
  Pstore              store,              /* handle to Store object          */
  Pint                *err_ind,           /* OUT error indicator             */
  Pcolr_map_rep       **rep               /* OUT colour mapping              */
                                          /* representation                  */

);
```

NOTE — The memory referenced by `*rep` is managed by `store`.

INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS

```
void pinq_ws_st_table_length_plus (

  Pint                      ws_type,      /* workstation type                */
  Pint                      *err_ind,     /* OUT error indicator             */
  Pws_st_tables_plus        *lengths      /* OUT lengths of workstation      */
                                          /* tables                          */

);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex A

(infomative)

Data types in compilation order and external functions

A.1 Macro definitions

Clause A.1, add the following macro definitions after the macro definition for PE_BAD_COLR*:*

```
/* Output Attribute Errors */
#define PE_BAD_DEPTH_CUE_MODE              (119) /* Ignoring function, the spec-
                                                    ified depth cue mode is not
                                                    available on the workstation
                                                    */
#define PE_DEPTH_CUE_INDX_LT_1             (120) /* Ignoring function, the
                                                    depth cue index is less
                                                    than one */
#define PE_COLR_MAP_INDX_LT_1              (121) /* Ignoring function, the
                                                    colour mapping index is
                                                    less than one */
#define PE_BAD_LINE_SHAD_METHOD            (122) /* Ignoring function, the spec-
                                                    ified polyline shading
                                                    method is not available on
                                                    the workstation */
#define PE_BAD_INT_SHAD_METHOD             (123) /* Ignoring function, the spec-
                                                    ified interior shading
                                                    method is not available on
                                                    the workstation */
#define PE_BAD_REFL_MODEL                  (124) /* Ignoring function, the spec-
                                                    ified reflectance model is
                                                    not available on the work-
                                                    station */
#define PE_TOTAL_COLR_RANGE_FIELDS_TOO_LARGE  (125)
                                                  /* Ignoring function, the
                                                     total of the colour range
                                                     fields in all the table
                                                     entries is too large */
#define PE_BAD_COLR_MAP_METHOD             (126) /* Ignoring function, the spec-
                                                    ified colour mapping method
                                                    is not available on the
                                                    specified workstation */
#define PE_BAD_APPROX_CRIT_TYPE            (127) /* Ignoring function, the spec-
                                                    ified approximation crite-
                                                    ria type is not available
                                                    on the specified workstation
                                                    */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PE_BAD_PARAM_SURF_CHAR_TYPE      (128) /* Ignoring function, the spec-
                                                  ified parametric surface
                                                  characteristics type is not
                                                  available on the specified
                                                  workstation */
#define PE_LIGHT_SOURCE_INDX_LT_1        (129) /* Ignoring function, the
                                                  light source index is less
                                                  than one */
#define PE_INVALID_REF_PLANES            (130) /* Ignoring function, invalid
                                                  reference planes; DQMIN >
                                                  DQMAX */
#define PE_BAD_LIGHT_SOURCE_TYPE         (131) /* Ignoring function, the spec-
                                                  ified light source type is
                                                  not available on the work-
                                                  station */
#define PE_SPOTLIGHT_SPREAD_ANGLE_RANGE  (132) /* Ignoring function, the spec-
                                                  ified spot light spread
                                                  angle is out of range */
#define PE_BAD_ACTIVATION_LIST           (133) /* Ignoring function, one of
                                                  the entries in the activa-
                                                  tion list or deactivation
                                                  list is less than one */
#define PE_GEN_COLR_NOT_INDIRECT         (134) /* Ignoring function, the
                                                  requested entry contains a
                                                  general colour specifica-
                                                  tion with colour type other
                                                  than INDIRECT */
#define PE_SAME_ACTIVATION_DEACTIVATION_ENTRY (135)
                                               /* Ignoring function, the same
                                                  entry exists in both the
                                                  activation and the deactiva-
                                                  tion list */
#define PE_COLR_COMPONENT_RANGE          (136) /* Ignoring function, one of
                                                  the components of the
                                                  colour specification is out
                                                  of range */
#define PE_BAD_DATA_MAP_METHOD           (137) /* Ignoring function, the spec-
                                                  ified data mapping method
                                                  is not available on the
                                                  specified workstation */
#define PE_BAD_DATA_RECORD_FIELD         (138) /* Ignoring function, one or
                                                  more of the fields within
                                                  the specified data record
                                                  is in error */
#define PE_BAD_REFL_PROP_TYPE            (139) /* Ignoring function, the spec-
                                                  ified reflectance property
                                                  type is not available on
                                                  the specified workstation */
#define PE_INVALID_DEPTH_CUE_SCALE       (140) /* Ignoring function, one of
                                                  the depth cue scale factors
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564200 819 ■


```
                                        is not in the required range
                                        */
```

Clause A.1, add the following macro definitions after the macro definition for PE_BAD_ERROR_FILE:

```
/* Output Primitive Errors */
#define PE_ORDER_LT_1                       (500) /* Ignoring function, the
                                                     order of a spline is less
                                                     than one */
#define PE_INSUFF_CTRL_POINTS_FOR_ORDER     (501) /* Ignoring function, a spline
                                                     does not contain enough con-
                                                     trol points for its speci-
                                                     fied order */
#define PE_ORDER_INCONSISTENT_KNOTS_CTRL_POINTS   (502)
                                                  /* Ignoring function, the
                                                     order of a spline is incon-
                                                     sistent with number of
                                                     knots and control points */
#define PE_KNOT_SEQUENCE_NOT_NONDECREASING  (503)
                                                  /* Ignoring function, the knot
                                                     sequence for a spline is
                                                     not non-decreasing */
#define PE_VERTEX_INDX_OUT_OF_RANGE         (504) /* Ignoring function, one or
                                                     more of the vertex indices
                                                     is out of range */
#define PE_PARAM_RANGE_INCONSISTENT_KNOTS   (506)
                                                  /* Ignoring function, the
                                                     parameter range for a
                                                     spline is inconsistent with
                                                     its knots */
#define PE_W_VALUE_LE_0                     (507) /* Ignoring function, the
                                                     fourth coordinate of a
                                                     rational control point is
                                                     less than or equal to zero */
#define PE_TRIM_CURVE_ORDER_LT_2            (508) /* Ignoring function, a trim-
                                                     ming curve's order is less
                                                     than two */
#define PE_INCONSISTENT_EDGE_FLAG_SPEC      (513) /* Ignoring function, the num-
                                                     ber of edge visibility
                                                     flags is inconsistent with
                                                     the number of edges in the
                                                     primitive */
#define PE_DATA_LIST_LENGTH_NOT_SAME        (514) /* Ignoring function, the data
                                                     lists do not all contain
                                                     the same number of entries */
#define PE_INCONSISTENT_FACET_SPEC          (516) /* Ignoring function, the
                                                     facet data is inconsistent
                                                     with the number of facets
                                                     in the primitive */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_PARAM_RANGE_COLR_SPLINE      (517) /* Ignoring function, the
                                                 parameter range of the
                                                 colour spline does not
                                                 include the effective param-
                                                 eter range of the geometry
                                                 spline */
#define PE_PARAM_RANGE_DATA_SPLINE      (518) /* Ignoring function, the
                                                 parameter range of the data
                                                 spline does not include the
                                                 effective parameter range
                                                 of the geometry spline */
#define PE_BAD_VERTEX_DATA_TYPE         (2207)/* Ignoring function, two or
                                                 more vertices do not con-
                                                 tain the same type of data*/
```

Clause A.1, add the following macro definitions after the macro definition for `Pfn_set_err_hand`*:*

```
#define Pfn_polyline_set3_colr                (183)
#define Pfn_fill_area_set3_data               (184)
#define Pfn_fill_area_set_data                (185)
#define Pfn_cell_array3_plus                  (186)
#define Pfn_set_of_fill_area_sets3_data       (187)
#define Pfn_set_of_fill_area_sets_data        (188)
#define Pfn_tri_set3_data                     (189)
#define Pfn_tri_set_data                      (190)
#define Pfn_tri_strip3_data                   (191)
#define Pfn_tri_strip_data                    (192)
#define Pfn_quad_mesh3_data                   (193)
#define Pfn_quad_mesh_data                    (194)
#define Pfn_non_uniform_b_spline_curve3       (195)
#define Pfn_non_uniform_b_spline_curve3_colr  (196)
#define Pfn_non_uniform_b_spline_surf3        (197)
#define Pfn_non_uniform_b_spline_surf3_data   (198)
#define Pfn_set_data_map_ind                  (199)
#define Pfn_set_refl_ind                      (200)
#define Pfn_set_back_int_ind                  (201)
#define Pfn_set_back_data_map_ind             (202)
#define Pfn_set_back_refl_ind                 (203)
#define Pfn_set_param_surf_ind                (204)
#define Pfn_set_line_colr                     (205)
#define Pfn_set_line_shad_method              (206)
#define Pfn_set_marker_colr                   (207)
#define Pfn_set_text_colr                     (208)
#define Pfn_set_facet_disting_mode            (209)
#define Pfn_set_facet_cull_mode               (210)
#define Pfn_set_int_colr                      (211)
#define Pfn_set_int_shad_method               (212)
#define Pfn_set_data_map_method               (213)
#define Pfn_set_refl_props                    (214)
#define Pfn_set_refl_model                    (215)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define Pfn_set_back_int_style                (216)
#define Pfn_set_back_int_style_ind            (217)
#define Pfn_set_back_int_colr                 (218)
#define Pfn_set_back_int_shad_method          (219)
#define Pfn_set_back_data_map_method          (220)
#define Pfn_set_back_refl_props               (221)
#define Pfn_set_back_refl_model               (222)
#define Pfn_set_light_source_st               (223)
#define Pfn_set_edge_colr                     (224)
#define Pfn_set_curve_approx_crit             (225)
#define Pfn_set_surf_approx_crit              (226)
#define Pfn_set_param_surf_chars              (227)
#define Pfn_set_rend_colr_model               (228)
#define Pfn_set_depth_cue_ind                 (229)
#define Pfn_set_colr_map_ind                  (230)
#define Pfn_set_line_rep_plus                 (231)
#define Pfn_set_marker_rep_plus               (232)
#define Pfn_set_text_rep_plus                 (233)
#define Pfn_set_int_rep_plus                  (234)
#define Pfn_set_edge_rep_plus                 (235)
#define Pfn_set_data_map_rep                  (236)
#define Pfn_set_refl_rep                      (237)
#define Pfn_set_param_surf_rep                (238)
#define Pfn_set_pat_rep_plus                  (239)
#define Pfn_set_light_source_rep              (240)
#define Pfn_set_depth_cue_rep                 (241)
#define Pfn_set_colr_map_rep                  (242)
```

Clause A.1, add the following macro definitions after the macro definition for `PMODEL_HLS`*:*

```
#define PCOLR_MAP_METHOD_TRUE                 (1)
#define PCOLR_MAP_METHOD_PSEUDO               (2)
#define PCOLR_MAP_METHOD_PSEUDO_N             (3)

#define PCOLR_INDIRECT                        (0)
#define PCOLR_RGB                             (1)
#define PCOLR_CIELUV                          (2)
#define PCOLR_HSV                             (3)
#define PCOLR_HLS                             (4)

#define PCURVE_APPROX_WS_DEP                  (1)
#define PCURVE_APPROX_CONST_PARAM_SUBD        (2)
#define PCURVE_APPROX_CHORDAL_SIZE_WC         (3)
#define PCURVE_APPROX_CHORDAL_SIZE_NPC        (4)
#define PCURVE_APPROX_CHORDAL_SIZE_DC         (5)
#define PCURVE_APPROX_CHORDAL_DEV_WC          (6)
#define PCURVE_APPROX_CHORDAL_DEV_NPC         (7)
#define PCURVE_APPROX_CHORDAL_DEV_DC          (8)
#define PCURVE_APPROX_REL_WC                  (9)
#define PCURVE_APPROX_REL_NPC                 (10)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define PCURVE_APPROX_REL_DC                    (11)

#define PDATA_MAP_METHOD_COLR                   (1)
#define PDATA_MAP_METHOD_SINGLE_UNIFORM         (2)
#define PDATA_MAP_METHOD_SINGLE_NON_UNIFORM     (3)
#define PDATA_MAP_METHOD_BI_UNIFORM             (4)
#define PDATA_MAP_METHOD_BI_NON_UNIFORM         (5)

#define PINT_SHAD_METHOD_NONE                   (1)
#define PINT_SHAD_METHOD_COLR                   (2)
#define PINT_SHAD_METHOD_DATA                   (3)
#define PINT_SHAD_METHOD_DATA_DOT               (4)
#define PINT_SHAD_METHOD_DATA_NORM              (5)

#define PLIGHT_SOURCE_AMB                       (1)
#define PLIGHT_SOURCE_DIR                       (2)
#define PLIGHT_SOURCE_POS                       (3)
#define PLIGHT_SOURCE_SPOT                      (4)

#define PPARAM_SURF_CHARS_NONE                  (1)
#define PPARAM_SURF_CHARS_WS_DEP                (2)
#define PPARAM_SURF_CHARS_ISOPARAM              (3)
#define PPARAM_SURF_CHARS_LEVEL_MC              (4)
#define PPARAM_SURF_CHARS_LEVEL_WC              (5)

#define PLINE_SHAD_METHOD_NONE                  (1)
#define PLINE_SHAD_METHOD_COLR                  (2)

#define PREFL_MODEL_NO_REFL                     (1)
#define PREFL_MODEL_AMB_REFL                    (2)
#define PREFL_MODEL_AMB_DIFF_REFL               (3)
#define PREFL_MODEL_AMB_DIFF_SPEC_REFL          (4)

#define PREFL_PROPS_SIMPLE_REFL                 (1)

#define PREND_COLR_MODEL_WS_DEP                 (0)
#define PREND_COLR_MODEL_RGB                    (1)
#define PREND_COLR_MODEL_CIELUV                 (2)
#define PREND_COLR_MODEL_HSV                    (3)
#define PREND_COLR_MODEL_HLS                    (4)

#define PSURF_APPROX_WS_DEP                     (1)
#define PSURF_APPROX_CONST_PARAM_SUBD           (2)
#define PSURF_APPROX_CHORDAL_SIZE_WC            (3)
#define PSURF_APPROX_CHORDAL_SIZE_NPC           (4)
#define PSURF_APPROX_CHORDAL_SIZE_DC            (5)
#define PSURF_APPROX_PLANAR_DEV_WC              (6)
#define PSURF_APPROX_PLANAR_DEV_NPC             (7)
#define PSURF_APPROX_PLANAR_DEV_DC              (8)
#define PSURF_APPROX_REL_WC                     (9)
#define PSURF_APPROX_REL_NPC                    (10)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564204 464 ■


```
#define PSURF_APPROX_REL_DC                    (11)
```

Clause A.1, add the following macro definitions after the macro definition for PLAST_PHIGS_ELEM*:*

```
#define PFIRST_PHIGS_PLUS_ELEM                 (PELEM_POLYLINE_SET3_COLR)
#define PLAST_PHIGS_PLUS_ELEM                  (PELEM_COLR_MAP_IND)
```

A.2 Types in compilation order

Clause A.2, modify the following PHIGS type definitions by adding the specified PHIGS PLUS enumerations:

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PASPECT_LINE_COLR,
  PASPECT_MARKER_COLR,
  PASPECT_TEXT_COLR,
  PASPECT_INT_COLR,
  PASPECT_EDGE_COLR,
  PASPECT_LINE_SHAD_METHOD,
  PASPECT_INT_SHAD_METHOD,
  PASPECT_DATA_MAP_METHOD,
  PASPECT_REFL_PROPS,
  PASPECT_REFL_MODEL,
  PASPECT_BACK_INT_STYLE,
  PASPECT_BACK_INT_STYLE_IND,
  PASPECT_BACK_INT_COLR,
  PASPECT_BACK_INT_SHAD_METHOD,
  PASPECT_BACK_DATA_MAP_METHOD,
  PASPECT_BACK_REFL_PROPS,
  PASPECT_BACK_REFL_MODEL,
  PASPECT_CURVE_APPROX_CRIT,
  PASPECT_SURF_APPROX_CRIT,
  PASPECT_PARAM_SURF_CHARS
  /* end of PHIGS PLUS enumerations */
} Paspect;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PATTR_REFL,
  PATTR_PARAM_SURF
  /* end of PHIGS PLUS enumerations */
} Pattrs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
  /* start of PHIGS enumerations */
  ...,
  /* end of PHIGS enumerations */
  /* start of PHIGS PLUS enumerations */
  PELEM_POLYLINE_SET3_COLR,
  PELEM_FILL_AREA_SET3_DATA,
  PELEM_FILL_AREA_SET_DATA,
  PELEM_CELL_ARRAY3_PLUS,
  PELEM_SET_OF_FILL_AREA_SETS3_DATA,
  PELEM_SET_OF_FILL_AREA_SETS_DATA,
  PELEM_TRI_SET3_DATA,
  PELEM_TRI_SET_DATA,
  PELEM_TRI_STRIP3_DATA,
  PELEM_TRI_STRIP_DATA,
  PELEM_QUAD_MESH3_DATA,
  PELEM_QUAD_MESH_DATA,
  PELEM_NON_UNIFORM_B_SPLINE_CURVE3,
  PELEM_NON_UNIFORM_B_SPLINE_CURVE3_COLR,
  PELEM_NON_UNIFORM_B_SPLINE_SURF3,
  PELEM_NON_UNIFORM_B_SPLINE_SURF3_DATA,
  PELEM_DATA_MAP_IND,
  PELEM_REFL_IND,
  PELEM_BACK_INT_IND,
  PELEM_BACK_DATA_MAP_IND,
  PELEM_BACK_REFL_IND,
  PELEM_PARAM_SURF_IND,
  PELEM_LINE_COLR,
  PELEM_LINE_SHAD_METHOD,
  PELEM_MARKER_COLR,
  PELEM_TEXT_COLR,
  PELEM_FACET_DISTING_MODE,
  PELEM_FACET_CULL_MODE,
  PELEM_INT_COLR,
  PELEM_INT_SHAD_METHOD,
  PELEM_DATA_MAP_METHOD,
  PELEM_REFL_PROPS,
  PELEM_REFL_MODEL,
  PELEM_BACK_INT_STYLE,
  PELEM_BACK_INT_STYLE_IND,
  PELEM_BACK_INT_COLR,
  PELEM_BACK_INT_SHAD_METHOD,
  PELEM_BACK_DATA_MAP_METHOD,
  PELEM_BACK_REFL_PROPS,
  PELEM_BACK_REFL_MODEL,
  PELEM_LIGHT_SOURCE_ST,
  PELEM_EDGE_COLR,
  PELEM_CURVE_APPROX_CRIT,
  PELEM_SURF_APPROX_CRIT,
  PELEM_PARAM_SURF_CHARS,
  PELEM_REND_COLR_MODEL,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  PELEM_DEPTH_CUE_IND,
  PELEM_COLR_MAP_IND
  /* end of PHIGS PLUS enumerations */
} Pelem_type;
```

Clause A.2, add the following type definitions after the type definition for `Pattrs`*:*

```
typedef enum {
  PDEPTH_CUE_SUPPR,
  PDEPTH_CUE_ALLOWED
} Pdepth_cue_mode;


typedef enum {
  PFACET_CULL_NONE,
  PFACET_CULL_BACKFACING,
  PFACET_CULL_FRONTFACING
} Pfacet_cull_mode;


typedef enum {
  PFACET_DISTING_OFF,
  PFACET_DISTING_ON
} Pfacet_disting_mode;


typedef enum {
  PPLACE_UNIFORM_OVER_SURF,
  PPLACE_UNIFORM_BETWEEN_KNOTS
} Pplacement;


typedef enum {
  PB_SPLINE_RATIONAL,
  PB_SPLINE_NON_RATIONAL
} Prationality;


typedef enum {
  PSOURCE_SELECT_COLR_ASPECT,
  PSOURCE_SELECT_VERT_COLR,
  PSOURCE_SELECT_VERT_DATA,
  PSOURCE_SELECT_FACET_COLR,
  PSOURCE_SELECT_FACET_DATA
} Psource_select;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
  PVIS_OFF,
  PVIS_ON
} Pvis_flag;
```

Clause A.2, add the following PHIGS PLUS types after the PHIGS type `Pescape_in_data`*:*

```
typedef struct {
  Pint              num_floats;          /* number of floats in list       */
  Pfloat            *floats;             /* list of floats                 */
} Pfloat_list;


typedef struct {
  Pint              num_lists;           /* number of float lists          */
  Pfloat_list       *floats;             /* list of float lists            */
} Pfloat_set;


typedef union {
  Prgb              *rgb;                /* pointer to RGB colour values   */
  Pcieluv           *cieluv;             /* pointer to CIELUV colour values*/
  Phls              *hls;                /* pointer to HLS colour values   */
  Phsv              *hsv;                /* pointer to HSV colour values   */
  Pdata             *unsupp;             /* pointer to unsupported colour  */
                                         /* values                         */
  int               impl_dep;            /* implementation defined         */
} Pcolr_rep_ptr;


typedef union {
  Pint              *colr_inds;          /* pointer to colour indices      */
  Pcolr_rep_ptr     colr_reps;           /* pointer to colour              */
                                         /* representations                */
} Pcolrv_ptr;


typedef struct {
  Pint              num_colrs;           /* number of colours              */
  Pcolrv_ptr        colrs;               /* list of colour values          */
} Pcolrv_list;


typedef struct {
  Pint              num_lists;           /* number of colour value lists   */
  Pcolrv_list       *colrs;              /* list of colour value lists     */
} Pcolrv_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint          method;                   /* colour mapping method          */
  union Pcolr_map_method_data {
    Pint        num_true_colrs;           /* number of true colours         */
    Pint        max_num_pseudo_colrs;     /* maximum number of pseudo       */
                                          /* colours                        */
    int         impl_dep;                 /* implementation defined         */
  } data;
} Pcolr_map_method;


typedef struct {
  Pint              method;               /* colour mapping method          */
  union Pcolr_map_rep_data {
    struct Pcolr_map_rep_pseudo {
      Pint          model;                /* colour model                   */
      Pfloat_list   weight_vector;        /* list of weight vector          */
      Pcolrv_list   colrs;                /* list of colour values          */
    } pseudo;
    struct Pcolr_map_rep_pseudo_N {
      Pint          model;                /* colour model                   */
      Pfloat_set    colrs;                /* list of lists of colour        */
                                          /* coordinates                    */
    } pseudo_N;
    int         impl_dep;                 /* implementation defined         */
  } data;
} Pcolr_map_rep;


typedef struct {
  Pint          method;                   /* colour mapping method          */
  union Pcolr_map_st_data {
    Pint        num_true_colrs;           /* number of available true       */
                                          /* colours                        */
    Pint        max_num_pseudo_colrs;     /* maximum number of pseudo       */
                                          /* colour entries                 */
    int         impl_dep;                 /* implementation defined         */
  } data;
} Pcolr_map_st;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint              type;                 /* curve approximation criteria   */
                                          /* type                           */
  union Pcurve_approx_crit_data {
    struct Pcurve_approx_const_param_subd {
      Pint          count;                /* count                          */
    } const_param_subd;
    struct Pcurve_approx_chordal_size_wc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_wc;
    struct Pcurve_approx_chordal_size_npc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_npc;
    struct Pcurve_approx_chordal_size_dc {
      Pfloat        approx;               /* approximation value            */
    } chordal_size_dc;
    struct Pcurve_approx_chordal_dev_wc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_wc;
    struct Pcurve_approx_chordal_dev_npc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_npc;
    struct Pcurve_approx_chordal_dev_dc {
      Pfloat        approx;               /* approximation value            */
    } chordal_dev_dc;
    struct Pcurve_approx_rel_wc {
      Pfloat        approx;               /* approximation value            */
    } rel_wc;
    struct Pcurve_approx_rel_npc {
      Pfloat        approx;               /* approximation value            */
    } rel_npc;
    struct Pcurve_approx_rel_dc {
      Pfloat        approx;               /* approximation value            */
    } rel_dc;
    Pdata           unsupp;               /* unsupported curve approximation*/
                                          /* criteria data                  */
    int             impl_dep;             /* implementation defined         */
  } data;
} Pcurve_approx_crit;


typedef struct {
  Pint              num_selectors;        /* number of selectors            */
  Psource_select    *selectors;           /* list of source selectors       */
} Psource_select_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint                    method;          /* data mapping method          */
  Psource_select_list     selectors;       /* list of source selectors     */
  union Pdata_map_rep_data {
    struct Pdata_map_rep_single_uniform {
      Pint                ind;             /* data value index             */
      Pfloat              lower_limit;     /* lower range limit            */
      Pfloat              upper_limit;     /* upper range limit            */
      Pint                colr_type;       /* colour type                  */
      Pcolrv_list         colr_values;     /* list of colour values        */
    } single_uniform;
    struct Pdata_map_rep_single_non_uniform {
      Pint                ind;             /* data value index             */
      Pfloat_list         range;           /* range boundaries             */
      Pint                colr_type;       /* colour type                  */
      Pcolrv_list         colr_values;     /* list of colour values        */
    } single_non_uniform;
    struct Pdata_map_rep_bi_uniform {
      Pint                inds[2];         /* data value indices           */
      Pfloat              lower_limit_a;   /* lower limit of Ra range      */
      Pfloat              upper_limit_a;   /* upper limit of Ra range      */
      Pfloat              lower_limit_b;   /* lower limit of Rb range      */
      Pfloat              upper_limit_b;   /* upper limit of Rb range      */
      Pint                colr_type;       /* colour type                  */
      Pcolrv_set          colr_values;     /* list of colour value lists   */
    } bi_uniform;
    struct Pdata_map_rep_bi_non_uniform {
      Pint                inds[2];         /* data value indices           */
      Pfloat_list         range_a;         /* Ra range boundaries          */
      Pfloat_set          range_b;         /* Rb range boundaries          */
      Pint                colr_type;       /* colour type                  */
      Pcolrv_set          colr_values;     /* array of colour values       */
    } bi_non_uniform;
    int                   impl_dep;        /* implementation defined       */
  } data;
} Pdata_map_rep;


typedef struct {
  Pfloat          cieluv_x;                /* x coefficient                */
  Pfloat          cieluv_y;                /* y coefficient                */
  Pfloat          cieluv_y_lum;            /* y luminance                  */
  Pfloat          w;                       /* homogeneous coordinate       */
} Phomo_cieluv;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pfloat            hue;                /* hue                          */
  Pfloat            lightness;          /* lightness                    */
  Pfloat            satur;              /* saturation                   */
  Pfloat            w;                  /* homogeneous coordinate       */
} Phomo_hls;


typedef struct {
  Pfloat            hue;                /* hue                          */
  Pfloat            satur;              /* saturation                   */
  Pfloat            value;              /* value                        */
  Pfloat            w;                  /* homogeneous coordinate       */
} Phomo_hsv;


typedef struct {
  Pfloat            red;                /* red                          */
  Pfloat            green;              /* green                        */
  Pfloat            blue;               /* blue                         */
  Pfloat            w;                  /* homogeneous coordinate       */
} Phomo_rgb;


typedef union {
  Phomo_rgb         *rgb;               /* pointer to homogeneous rgb   */
                                        /* colour value                 */
  Phomo_cieluv      *cieluv;            /* pointer to homogeneous cieluv */
                                        /* colour value                 */
  Phomo_hls         *hls;               /* pointer to homogeneous hls   */
                                        /* colour value                 */
  Phomo_hsv         *hsv;               /* pointer to homogeneous hsv   */
                                        /* colour value                 */
  int               impl_dep;           /* implementation defined       */
} Phomo_colr_rep_ptr;


typedef union {
  Pint                  colr_ind;       /* colour index                 */
  Pcolr_rep             colr_rep;       /* colour representation        */
} Pcolrv;


typedef struct {
  Pint              colr_type;          /* colour type                  */
  Pcolrv            colr_value;         /* colour value                 */
} Pgcolr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint              type;                 /* light source type              */
  union Plight_source_rep_data {
    struct Plight_source_rep_amb {
      Pgcolr        colr;                 /* light source colour            */
    } amb;
    struct Plight_source_rep_dir {
      Pgcolr        colr;                 /* light source colour            */
      Pvec3         dir;                  /* light source direction         */
    } dir;
    struct Plight_source_rep_pos {
      Pgcolr        colr;                 /* light source colour            */
      Ppoint3       pos;                  /* light source position          */
      Pfloat        c1;                   /* light source attenuation       */
                                          /* coefficient                    */
      Pfloat        c2;                   /* light source attenuation       */
                                          /* coefficient                    */
    } pos;
    struct Plight_source_rep_spot {
      Pgcolr        colr;                 /* light source colour            */
      Ppoint3       pos;                  /* light source position          */
      Pvec3         dir;                  /* light source direction         */
      Pfloat        concent_exp;          /* concentration exponent         */
      Pfloat        c1;                   /* light source attenuation       */
                                          /* coefficient                    */
      Pfloat        c2;                   /* light source attenuation       */
                                          /* coefficient                    */
      Pfloat        spread_angle;         /* light source spread angle      */
    } spot;
    int             impl_dep;             /* implementation defined         */
  } data;
} Plight_source_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint              type;                 /* parametric surface             */
                                          /* characteristics type           */
  union Pparam_surf_chars_data {
    struct Pparam_surf_isoparam {
      Pplacement    placement;            /* curve placement: UNIFORM OVER  */
                                          /* SURFACE, UNIFORM BETWEEN KNOTS */
      Pint          num_u_curves;         /* curve count in u direction     */
      Pint          num_v_curves;         /* curve count in v direction     */
    } isoparam;
    struct Pparam_surf_level_mc {
      Ppoint3       origin;               /* origin point                   */
      Pvec3         dir;                  /* direction vector               */
      Pfloat_list   params;               /* parameter list                 */
    } level_mc;
    struct Pparam_surf_level_wc {
      Ppoint3       origin;               /* origin point                   */
      Pvec3         dir;                  /* direction vector               */
      Pfloat_list   params;               /* parameter list                 */
    } level_wc;
    Pdata           unsupp;               /* unsupported parametric         */
                                          /* surface characteristics data   */
    int             impl_dep;             /* implementation defined         */
  } data;
} Pparam_surf_chars;


typedef struct {
  Pint              type;                 /* reflectance property type      */
  union Prefl_prop_data {
    struct Prefl_prop_simple_refl {
      Pfloat        amb_coeff;            /* ambient reflection             */
                                          /* coefficient                    */
      Pfloat        diff_coeff;           /* diffuse reflection             */
                                          /* coefficient                    */
      Pfloat        spec_coeff;           /* specular reflection            */
                                          /* coefficient                    */
      Pgcolr        spec_colr;            /* specular colour                */
      Pfloat        spec_exp;             /* specular exponent              */
    } simple_refl;
    Pdata           unsupp;               /* unsupported reflectance        */
                                          /* property data                  */
    int             impl_dep;             /* implementation defined         */
  } data;
} Prefl_prop;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint              type;                  /* surface approximation criteria */
                                           /* type                           */
  union Psurf_approx_crit_data {
    struct Psurf_approx_const_param_subd {
      Pint          u_count;               /* u count                        */
      Pint          v_count;               /* v count                        */
    } const_param_subd;
    struct Psurf_approx_chordal_size_wc {
      Pfloat        u_approx;              /* u approximation value          */
      Pfloat        v_approx;              /* v approximation value          */
    } chordal_size_wc;
    struct Psurf_approx_chordal_size_npc {
      Pfloat        u_approx;              /* u approximation value          */
      Pfloat        v_approx;              /* v approximation value          */
    } chordal_size_npc;
    struct Psurf_approx_chordal_aize_dc {
      Pfloat        u_approx;              /* u approximation value          */
      Pfloat        v_approx;              /* v approximation value          */
    } chordal_size_dc;
    struct Psurf_approx_planar_dev_wc {
      Pfloat        approx;                /* approximation value            */
    } planar_dev_wc;
    struct Psurf_approx_planar_dev_npc {
      Pfloat        approx;                /* approximation value            */
    } planar_dev_npc;
    struct Psurf_approx_planar_dev_dc {
      Pfloat        approx;                /* approximation value            */
    } planar_dev_dc;
    struct Psurf_approx_rel_wc {
      Pfloat        approx;                /* approximation value            */
    } rel_wc;
    struct Psurf_approx_rel_npc {
      Pfloat        approx;                /* approximation value            */
    } rel_npc;
    struct Psurf_approx_rel_dc {
      Pfloat        approx;                /* approximation value            */
    } rel_dc;
    Pdata           unsupp;                /* unsupported surface            */
                                           /* approximation criteria data    */
    int             impl_dep;              /* implementation defined         */
  } data;
} Psurf_approx_crit;


typedef union {
  Pcolr_rep_ptr         points;            /* pointer to colour points       */
  Phomo_colr_rep_ptr    homo_points;       /* pointer to homogeneous colour  */
                                           /* points                         */
} Pcolr_ctrl_point_ptr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564216 186 ■


```
typedef struct {
  Prationality          rationality;      /* control point rationality      */
  Pint_size             dims;             /* dimensions of control point    */
                                          /* array                          */
  Pcolr_ctrl_point_ptr  colr_points;      /* array of colour control points */
} Pcolr_ctrl_point_array;


typedef struct {
  Prationality          rationality;      /* control point rationality      */
  Pint                  num_points;       /* number of control points       */
  Pcolr_ctrl_point_ptr  points;           /* list of colour control points  */
} Pcolr_ctrl_point_list;


typedef struct {
  Pint_size         dims;                 /* colour value array dimensions  */
  Pcolrv_ptr        colrs;                /* array of colour values         */
} Pcolrv_array;


typedef struct {
  Pint              num_lists;            /* number of contour lists        */
  Pint_list         *contours;            /* list of contour lists          */
} Pcontour_set;


typedef struct {
  Pint              num_contour_sets;     /* number of contour sets         */
  Pcontour_set      *contours;            /* list of contour sets           */
} Pcontour_set_list;


typedef struct {
  Pfloat            wx;                   /* wx coordinate                  */
  Pfloat            wy;                   /* wy coordinate                  */
  Pfloat            wz;                   /* wz coordinate                  */
  Pfloat            w;                    /* homogeneous coordinate         */
} Phomo_point4;


typedef union {
  Ppoint3           *point3;              /* pointer to 3D non-homogeneous  */
                                          /* points                         */
  Phomo_point4      *homo_point4;         /* pointer to 4D homogeneous      */
                                          /* points                         */
} Ppoint3_ptr;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564217 012 ■


```
typedef struct {
  Prationality      rationality;          /* control point rationality      */
  Pint_size         dims;                 /* control point array u,v        */
                                          /* dimensions                     */
  Ppoint3_ptr       points;               /* array of control points        */
} Pctrl_point3_array;


typedef struct {
  Pfloat            wx;                   /* wx coordinate                  */
  Pfloat            wy;                   /* wy coordinate                  */
  Pfloat            w;                    /* homogeneous coordinate         */
} Phomo_point3;


typedef union {
  Ppoint            *point;               /* pointer to 2D non-homogeneous  */
                                          /* points                         */
  Phomo_point3      *homo_point3;         /* pointer to 3D homogeneous      */
                                          /* points                         */
} Ppoint_ptr;


typedef struct {
  Prationality      rationality;          /* control point rationality      */
  Pint              num_points;           /* number of control points       */
  Ppoint_ptr        points;               /* list of control points         */
} Pctrl_point_list;


typedef struct {
  Prationality      rationality;          /* control point rationality      */
  Pint              num_points;           /* number of control points       */
  Ppoint3_ptr       points;               /* list of control points         */
} Pctrl_point3_list;


typedef struct {
  Pint                      order;        /* spline order                   */
  Pfloat_list               knots;        /* list of knot values            */
  Pint                      colr_type;    /* colour type                    */
  Pcolr_ctrl_point_list     colrs;        /* list of colour control points  */
} Pcurve_colr_spline;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint                  order;          /* spline order                  */
  Pfloat_list           knots;          /* knots                         */
  Pfloat                low_limit;      /* lower parameter range limit   */
  Pfloat                high_limit;     /* high parameter range limit    */
  Pctrl_point3_list     ctrl_points;    /* list of control points        */
} Pcurve_geom_spline;


typedef struct {
  Prationality      rationality;        /* control point rationality     */
  Pint_size         dims;               /* control point array u,v       */
                                        /* dimensions                    */
  Pint              points_dims;        /* control point dimensionality  */
  Pfloat            *ctrl_points;       /* array of control points       */
} Pdata_ctrl_point_array;


typedef struct {
  Pint              num_modes;          /* number of depth cue modes     */
  Pdepth_cue_mode   *modes;             /* list of depth cue modes       */
} Pdepth_cue_mode_list;


typedef struct {
  Pdepth_cue_mode   mode;               /* depth cue mode                */
  Pfloat            ref_planes[2];      /* depth cue reference planes    */
                                        /* (min, max)
  Pfloat            scale[2];           /* depth cue scale factors       */
                                        /* (min, max)
  Pgcolr            gcolr;              /* depth cue colour              */
} Pdepth_cue_rep;


typedef struct {
  Pdyn_mod          data_map;           /* data mapping representation   */
  Pdyn_mod          refl;               /* reflectance representation    */
  Pdyn_mod          param_surf;         /* parametric surface            */
                                        /* representation                */
  Pdyn_mod          light_source;       /* light source representation   */
  Pdyn_mod          depth_cue;          /* depth cue representation      */
  Pdyn_mod          colr_map;           /* colour mapping representation */
} Pdyn_ws_attrs_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pedge_flag        flag;                /* edge flag                    */
  Pint              type;                /* edgetype                     */
  Pfloat            width;               /* edgewidth scale factor       */
  Pgcolr            colr;                /* edge colour                  */
} Pedge_bundle_plus;


typedef Pedge_flag Pedge_flag_pair[2];


typedef Pedge_flag Pedge_flag_triplet[3];


typedef struct {
  Pint              num_edge_flag_triplets;/* number of edge flag triplets */
  Pedge_flag_triplet*edge_flag_triplets;   /* list of edge flag triplets   */
} Pedge_flag_triplet_list;


typedef struct {
  Pint_size         dims;                /* edge flag array dimensions   */
  Pedge_flag_pair   *edge_flags;         /* array of edge flag pairs     */
} Pedge_flag_array;


typedef struct {
  Pint              num_edges;           /* number of edges in list      */
  Pedge_flag        *edge_flags;         /* list of edge flags           */
} Pedge_flag_list;


typedef struct {
  Pint              num_lists;           /* number of edge flag lists    */
  Pedge_flag_list   *edge_flags;         /* list of edge flag lists      */
} Pedge_flag_set;


typedef struct {
  Pint              num_edge_flag_sets;  /* number of edge flag sets     */
  Pedge_flag_set    *edge_flag_sets;     /* list of edge flag sets       */
} Pedge_flag_set_list;


typedef struct {
  Pint              num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv            *facet_colrv;        /* facet colour                 */
  Pvec3             *facet_norm;         /* facet normal                 */
  Pfloat            *facet_data;         /* list of facet data           */
} Pfacet;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564220 607 ■


```
typedef struct {
  Pint_size         dims;                /* facet array dimensions          */
  Pint              num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv_ptr        facet_colrvs;        /* array of facet colour values    */
  Pvec3             *facet_norms;        /* array of facet normals          */
  Pfloat            *facet_data;         /* array of facet data lists       */
} Pfacet_array;


typedef struct {
  Pint              num_facets;          /* number of facets                */
  Pint              num_data_per_facet;  /* number of data values per-facet*/
  Pcolrv_ptr        facet_colrvs;        /* list of facet colour values     */
  Pvec3             *facet_norms;        /* list of facet normals           */
  Pfloat            *facet_data;         /* list of facet data lists        */
} Pfacet_list;


typedef struct {
  Pint_style        style;               /* interior style                  */
  Pint              style_ind;           /* interior style index            */
  Pgcolr            colr;                /* interior colour                 */
  Pint              shad_method;         /* shading method                  */
} Pint_bundle_plus;


typedef struct {
  Pint              num_int_styles;      /* number of interior styles       */
  Pint_style        int_styles[5];       /* list of available interior      */
                                         /* styles                          */
  Pint              num_hatch_styles;    /* number of available hatch       */
                                         /* styles                          */
  Pint_list         hatch_styles;        /* list of available hatch styles  */
  Pint_list         shad_methods;        /* list of available shading       */
                                         /* methods                         */
  Pint              num_pred_int_inds;   /* number of predefined interior   */
                                         /* indices                         */
} Pint_facs_plus;


typedef struct {
  Pint              type;                /* line type                       */
  Pfloat            width;               /* linewidth scale factor          */
  Pgcolr            colr;                /* polyline colour                 */
  Pint              shad_method;         /* polyline shading method         */
  Pcurve_approx_crit approx_crit;        /* curve approximation criteria    */
                                         /* data record                     */
} Pline_bundle_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint              num_types;          /* number of available linetypes  */
  Pint_list         types;              /* list of line types             */
  Pint              num_widths;         /* number of available line widths*/
  Pfloat            nom_width;          /* nominal linewidth              */
  Pfloat            min_width;          /* minimum linewidth              */
  Pfloat            max_width;          /* maximum linewidth              */
  Pint_list         shad_methods;       /* list of available line shading */
                                        /* methods                        */
  Pint              num_inds;           /* number of predefined bundle    */
                                        /* indices                        */
} Pline_facs_plus;


typedef struct {
  Pint              type;               /* marker type                    */
  Pfloat            size;               /* marker size scale factor       */
  Pgcolr            colr;               /* polymarker colour              */
} Pmarker_bundle_plus;


typedef struct {
  Psurf_approx_crit     approx_crit;    /* surface approximation criteria */
                                        /* data                           */
  Pparam_surf_chars     chars_data;     /* parametric surface             */
                                        /* characteristics data           */
} Pparam_surf_rep;


typedef struct {
  Pint                  colr_type;      /* colour type                    */
  Pcolrv_array          colrs;          /* array of colour values         */
} Ppat_rep_plus;


typedef struct {
  Pint                refl_model;       /* reflectance model              */
  Prefl_prop          refl_prop;        /* reflectance property data      */
} Prefl_rep;


typedef struct {
  Pint              num_sets;           /* number of fill area sets       */
  Pint              num_data_per_facet; /* number of data values per facet*/
  Pcolrv_ptr        facet_colrvs;       /* list of facet colours          */
  Pvec3             *facet_norms;       /* list of facet normals          */
  Pfloat            *facet_data;        /* list of facet data lists       */
} Pset_of_fill_area_sets_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564222 48T ■


```
typedef struct {
  Pint                    u_order;       /* u spline order                 */
  Pint                    v_order;       /* v spline order                 */
  Pfloat_list             u_knots;       /* list of u knot values          */
  Pfloat_list             v_knots;       /* list of v knot values          */
  Pint                    colr_type;     /* colour type                    */
  Pcolr_ctrl_point_array  colrs;         /* array of colour control points */
} Psurf_colr_spline;


typedef struct {
  Pint                    u_order;       /* u spline order                 */
  Pint                    v_order;       /* v spline order                 */
  Pfloat_list             u_knots;       /* list of u knot values          */
  Pfloat_list             v_knots;       /* list of v knot values          */
  Pdata_ctrl_point_array  data_points;   /* array of data control          */
                                         /* points                         */
} Psurf_data_spline;


typedef struct {
  Pint                num_data_spline;   /* number of data splines         */
  Psurf_data_spline   *data_splines;     /* list of data splines           */
} Psurf_data_spline_list;


typedef struct {
  Pint                  u_order;         /* u spline order                 */
  Pint                  v_order;         /* v spline order                 */
  Pfloat_list           u_knots;         /* u knots                        */
  Pfloat_list           v_knots;         /* v knots                        */
  Pctrl_point3_array    ctrl_points;     /* array of control points        */
} Psurf_geom_spline;


typedef struct {
  Pint              font;                /* text font                      */
  Ptext_prec        precision;           /* text precision                 */
  Pfloat            char_expan;          /* character expansion factor     */
  Pfloat            char_space;          /* character spacing              */
  Pgcolr            colr;                /* text colour                    */
} Ptext_bundle_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint                num_sets;           /* number of triangle sets        */
  Pint                num_data_per_facet; /* number of data values per      */
                                          /* facet                          */
  Pcolrv_ptr          facet_colrvs;       /* list of facet colours          */
  Pvec3               *facet_norms;       /* list of facet normals          */
  Pfloat              *facet_data;        /* list of facet data lists       */
} Ptri_set_data;


typedef struct {
  Pcurve_approx_crit  approx_crit;        /* trimming curve approx.         */
                                          /* criteria data record           */
  Pvis_flag           visibility_flag;    /* curve visibility flag          */
  Pint                order;              /* curve order                    */
  Pfloat_list         knots;              /* curve knot vector              */
  Pfloat              low_limit;          /* curve parameter range lower    */
                                          /* limit                          */
  Pfloat              high_limit;         /* curve parameter range upper    */
                                          /* limit                          */
  Pctrl_point_list    ctrl_points;        /* curve control points           */
} Ptrim_curve;


typedef struct {
  Pint              num_trim_curves;      /* number of trimming curves      */
  Ptrim_curve       *trim_curves;         /* list of trimming curves forming*/
                                          /* a closed loop                  */
} Ptrim_curve_loop;


typedef struct {
  Pint                num_trim_loops;     /* number of trimming curve loops */
  Ptrim_curve_loop    *trim_loops;        /* list of trimming curve loops   */
} Ptrim_curve_loop_list;


typedef struct {
  Pint_size           dims;               /* dimensions of vertex array     */
  Pint                num_data_per_vertex;/* number of data values          */
                                          /* per-vertex                     */
  Ppoint              *vertex_points;     /* array of vertex points         */
  Pcolrv_ptr          vertex_colrvs;      /* array of vertex colours        */
  Pvec3               *vertex_norms;      /* array of vertex normals        */
  Pfloat              *vertex_data;       /* array of vertex data lists     */
} Pvertex_array;


typedef Pint Pvertex_ind_triplet[3];
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint        num_vertex_ind_triplets;     /* number of vertex index          */
                                           /* triplets                        */
  Pvertex_ind_triplets*vertex_ind_triplets;/* list of vertex index triplets   */
} Pvertex_ind_triplet_list;


typedef struct {
  Pint              num_vertices;          /* number of vertices              */
  Pint              num_data_per_vertex;   /* number of data values           */
                                           /* per-vertex                      */
  Ppoint            *vertex_points;        /* list of vertex points           */
  Pcolrv_ptr        vertex_colrvs;         /* list of vertex colours          */
  Pvec3             *vertex_norms;         /* list of vertex normals          */
  Pfloat            *vertex_data;          /* list of vertex data lists       */
} Pvertex_list;


typedef struct {
  Pint              num_lists;             /* number of vertex lists          */
  Pvertex_list      *vertices;             /* list of vertex lists            */
} Pvertex_set;


typedef struct {
  Pint_size         dims;                  /* dimensions of vertex array      */
  Pint              num_data_per_vertex;   /* number of data values           */
                                           /* per-vertex                      */
  Ppoint3           *vertex_points;        /* array of vertex points          */
  Pcolrv_ptr        vertex_colrvs;         /* array of vertex colours         */
  Pvec3             *vertex_norms;         /* array of vertex normals         */
  Pfloat            *vertex_data;          /* array of vertex data lists      */
} Pvertex3_array;


typedef struct {
  Pint              num_vertices;          /* number of vertices              */
  Pint              num_data_per_vertex;   /* number of data values           */
                                           /* per-vertex                      */
  Ppoint3           *vertex_points;        /* list of vertex points           */
  Pcolrv_ptr        vertex_colrvs;         /* list of vertex colours          */
  Pvec3             *vertex_norms;         /* list of vertex normals          */
  Pfloat            *vertex_data;          /* list of vertex data lists       */
} Pvertex3_list;


typedef struct {
  Pint              num_lists;             /* number of vertex lists          */
  Pvertex3_list     *vertices;             /* list of vertex lists            */
} Pvertex3_set;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint              data_map_rep;         /* max. # of data mapping table   */
                                          /* entries                        */
  Pint              refl_rep;             /* max. # of reflectance table    */
                                          /* entries                        */
  Pint              param_surf_rep;       /* max. # of param. surface table */
                                          /* entries                        */
  Pint              light_source_rep;     /* max. # of light source table   */
                                          /* entries                        */
  Pint              depth_cue_rep;        /* max. # of depth cue table      */
                                          /* entries                        */
  Pint              colr_map_rep;         /* max. # of colour mapping table */
                                          /* entries                        */
} Pws_st_tables_plus;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564226 025 ■

Clause A.2, modify the PHIGS `Pelem_data` *type definition by adding the specified PHIGS PLUS fields as defined below:*

```
typedef union {
  /* start of PHIGS element data */
  ...,
  /* end of PHIGS element data */
  /* start of PHIGS PLUS element data */
  struct Pelem_polyline_set3_colr {
    Pint            colr_type;          /* colour type                    */
    Pvertex3_set    vertex_data;        /* list of vertex lists           */
  } polyline_set3_colr;
  struct Pelem_fill_area_set3_data {
    Pint            colr_type;          /* colour type                    */
    Pfacet          *facet_data;        /* facet data                     */
    Pedge_flag_set  *edge_flags;        /* list of edge flag lists        */
    Pvertex3_set    vertex_data;        /* list of vertex lists           */
  } fill_area_set3_data;
  struct Pelem_fill_area_set_data {
    Pint            colr_type;          /* colour type                    */
    Pfacet          *facet_data;        /* facet data                     */
    Pedge_flag_set  *edge_flags;        /* list of edge flag lists        */
    Pvertex_set     vertex_data;        /* list of vertex lists           */
  } fill_area_set_data;
  struct Pelem_cell_array3_plus {
    Pparal          paral;              /* parallelogram P, Q, R          */
    Ppat_rep_plus   pattern;            /* pattern colour array           */
  } cell_array3_plus;
  struct Pelem_set_of_fill_area_sets3_data {
    Pint                colr_type;      /* colour type                    */
    Pset_of_fill_area_sets_data *set_data; /* fill area sets data         */
    Pedge_flag_set_list  *edge_flags;   /* list of edge flag sets         */
    Pvertex3_list        vertex_data;   /* list of vertices               */
    Pcontour_set_list    contours;      /* list of contour sets           */
  } set_of_fill_area_sets3_data;
  struct Pelem_set_of_fill_area_sets_data {
    Pint                colr_type;      /* colour type                    */
    Pset_of_fill_area_sets_data *set_data; /* fill area sets data         */
    Pedge_flag_set_list  *edge_flags;   /* list of edge flag sets         */
    Pvertex_list         vertex_data;   /* list of vertices               */
    Pcontour_set_list    contours;      /* list of contour sets           */
  } set_of_fill_area_sets_data;
  struct Pelem_tri_set3_data {
    Pint                    colr_type;  /* colour type                    */
    Ptri_set_data           *set_data;  /* triangle set data              */
    Pedge_flag_triplet_list *edge_flags; /* list of triangle edge flag    */
                                        /* triplets                       */
    Pvertex3_list           vertex_data; /* list of vertices              */
    Pvertex_ind_triplet_list triplets;  /* list of triangle vertex index  */
                                        /* triplets                       */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
} tri_set3_data;
struct Pelem_tri_set_data {
  Pint                    colr_type;     /* colour type                    */
  Ptri_set_data           *set_data;     /* triangle set data              */
  Pedge_flag_triplet_list *edge_flags;   /* list of triangle edge flag     */
                                         /* triplets                       */
  Pvertex_list            vertex_data;   /* list of vertices               */
  Pvertex_ind_triplet_list triplets;     /* list of triangle vertex index  */
                                         /* triplets                       */
} tri_set_data;
struct Pelem_tri_strip3_data {
  Pint                colr_type;         /* colour type                    */
  Pfacet_list         *facet_data;       /* list of facet data             */
  Pedge_flag_list     *edge_flags;       /* list of edge flags             */
  Pvertex3_list       vertex_data;       /* list of vertex data            */
} tri_strip3_data;
struct Pelem_tri_strip_data {
  Pint                colr_type;         /* colour type                    */
  Pfacet_list         *facet_data;       /* list of facet data             */
  Pedge_flag_list     *edge_flags;       /* list of edge flags             */
  Pvertex_list        vertex_data;       /* list of vertex data            */
} tri_strip_data;
struct Pelem_quad_mesh3_data {
  Pint                colr_type;         /* colour type                    */
  Pfacet_array        *facet_data;       /* array of facet data            */
  Pedge_flag_array    *edge_flags;       /* array of edge flags            */
  Pvertex3_array      vertex_data;       /* array of vertex data           */
} quad_mesh3_data;
struct Pelem_quad_mesh_data {
  Pint                colr_type;         /* colour type                    */
  Pfacet_array        *facet_data;       /* array of facet data            */
  Pedge_flag_array    *edge_flags;       /* array of edge flags            */
  Pvertex_array       vertex_data;       /* array of vertex data           */
} quad_mesh_data;
struct Pelem_non_uniform_b_spline_curve3 {
  Pcurve_geom_spline    geom_spline;     /* geometry spline                */
} non_uniform_b_spline_curve3;
struct Pelem_non_uniform_b_spline_curve3_colr {
  Pcurve_geom_spline    geom_spline;     /* geometry spline                */
  Pcurve_colr_spline    *colr_spline;    /* colour spline                  */
} non_uniform_b_spline_curve3_colr;
struct Pelem_non_uniform_b_spline_surf3 {
  Psurf_geom_spline     geom_spline;     /* geometry spline                */
  Ptrim_curve_loop_list *trim_loops;     /* list of trimming curve loops   */
} non_uniform_b_spline_surf3;
struct Pelem_non_uniform_b_spline_surf3_data {
  Psurf_geom_spline     geom_spline;     /* geometry spline                */
  Ptrim_curve_loop_list *trim_loops;     /* list of trimming curve loops   */
  Psurf_colr_spline     *colr_spline;    /* colour spline                  */
  Psurf_data_spline_list *data_spline;   /* data spline                    */
} non_uniform_b_spline_surf3_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
  Pgcolr                colr;                /* general colour                */
  Pfacet_disting_mode   facet_disting_mode;  /* facet distinguishing mode     */
  Pfacet_cull_mode      facet_cull_mode;     /* facet culling mode            */
  Pdata_map_rep         data_map_method;     /* data mapping method           */
  Prefl_prop            refl_prop;           /* reflectance property data     */
  struct Pelem_light_source_st {
    Pint_list              act_list;         /* activation list               */
    Pint_list              deact_list;       /* deactivation list             */
  } light_source_st;
  Pcurve_approx_crit  curve_approx_crit;     /* curve approximation criteria  */
                                             /* data record                   */
  Psurf_approx_crit   surf_approx_crit;      /* surface approximation criteria */
                                             /* data record                   */
  Pparam_surf_chars   param_surf_chars;      /* parametric surface            */
                                             /* characteristics data record   */
  /* end of PHIGS PLUS element data */
} Pelem_data;
```

A.3 External functions

Clause A.3, change the following PHIGS external functions:

```
/* INQUIRE GENERALIZED DRAWING PRIMITIVE 3 */
extern void pinq_gdp3 (
  Pint                    ws_type,           /* workstation type              */
  Pint                    gdp,               /* GDP function identifier       */
  Pint                    *err_ind,          /* OUT error indicator           */
  Pint                    *num_attr,         /* OUT number of attributes used */
  Pattrs                  attr[7]            /* OUT list of attributes used   */
);

/* INQUIRE GENERALIZED DRAWING PRIMITIVE */
extern void pinq_gdp (
  Pint                    ws_type,           /* workstation type              */
  Pint                    gdp,               /* GDP function identifier       */
  Pint                    *err_ind,          /* OUT error indicator           */
  Pint                    *num_attr,         /* OUT number of attributes used */
  Pattrs                  attr[7]            /* OUT list of attributes used   */
);
```

Clause A.3, add the following PHIGS PLUS external functions after PHIGS external functions:

```
/* POLYLINE SET 3 WITH COLOUR */
extern void ppolyline_set3_colr (
  Pint                    colr_type,         /* colour type                   */
  const Pvertex3_set      *vertex_data       /* list of vertex lists          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* FILL AREA SET 3 WITH DATA */
extern void pfill_area_set3_data (
  Pint                    colr_type,       /* colour type                    */
  const Pfacet            *facet_data,     /* facet data                     */
  const Pedge_flag_set    *edge_flags,     /* list of edge flag lists        */
  const Pvertex3_set      *vertex_data     /* list of vertex lists           */
);

/* FILL AREA SET WITH DATA */
extern void pfill_area_set_data (
  Pint                    colr_type,       /* colour type                    */
  const Pfacet            *facet_data,     /* facet data                     */
  const Pedge_flag_set    *edge_flags,     /* list of edge flag lists        */
  const Pvertex_set       *vertex_data     /* list of vertex lists           */
);

/* CELL ARRAY 3 PLUS */
extern void pcell_array3_plus (
  const Pparal            *paral,          /* parallelogram P, Q, R          */
  const Ppat_rep_plus     *pat             /* pattern colour array           */
);

/* SET OF FILL AREA SETS 3 WITH DATA */
extern void pset_of_fill_area_sets3_data (
  Pint                       colr_type,    /* colour type                    */
  const Pset_of_fill_area_sets_data  *set_data,/*fill area set data          */
  const Pedge_flag_set_list  *edge_flags,  /* list of edge flag sets         */
  const Pvertex3_list        *vertex_data, /* list of vertices               */
  const Pcontour_set_list    *contours     /* list of contour sets           */
);

/* SET OF FILL AREA SETS WITH DATA */
extern void pset_of_fill_area_sets_data (
  Pint                       colr_type,    /* colour type                    */
  const Pset_of_fill_area_sets_data  *set_data,/*fill area set data          */
  const Pedge_flag_set_list  *edge_flags,  /* list of edge flag sets         */
  const Pvertex_list         *vertex_data, /* list of vertices               */
  const Pcontour_set_list    *contours     /* list of contour sets           */
);

/* TRIANGLE SET 3 WITH DATA */
extern void ptri_set3_data (
  Pint                     colr_type,      /* colour type                    */
  const Ptri_set_data      *set_data,      /* triangle set data              */
  const Pedge_flag_triplet_list *edge_flags,/*list of triangle edge flag     */
                                           /* triplets                       */
  const Pvertex3_list      *vertex_data,   /* list of vertices               */
  const Pvertex_ind_triplet_list *triplets /* list of triangle vertex index  */
                                           /* triplets                       */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* TRIANGLE SET WITH DATA */
extern void ptri_set_data (
  Pint                        colr_type,      /* colour type                    */
  const Ptri_set_data         *set_data,      /* triangle set data              */
  const Pedge_flag_triplet_list *edge_flags,/*list of triangle edge flag      */
                                              /* triplets                       */
  const Pvertex_list          *vertex_data,   /* list of vertices               */
  const Pvertex_ind_triplet_list *triplets /* list of triangle vertex index  */
                                              /* triplets                       */
);

/* TRIANGLE STRIP 3 WITH DATA */
extern void ptri_strip3_data (
  Pint                      colr_type,        /* colour type                    */
  const Pfacet_list         *facet_data,      /* list of facet data             */
  const Pedge_flag_list     *edge_flags,      /* list of edge flags             */
  const Pvertex3_list       *vertex_data      /* list of vertex data            */
);

/* TRIANGLE STRIP WITH DATA */
extern void ptri_strip_data (
  Pint                      colr_type,        /* colour type                    */
  const Pfacet_list         *facet_data,      /* list of facet data             */
  const Pedge_flag_list     *edge_flags,      /* list of edge flags             */
  const Pvertex_list        *vertex_data      /* list of vertex data            */
);

/* QUADRILATERAL MESH 3 WITH DATA */
extern void pquad_mesh3_data (
  Pint                      colr_type,        /* colour type                    */
  const Pfacet_array        *facet_data,      /* array of facet data            */
  const Pedge_flag_array    *edge_flags,      /* array of edge flags            */
  const Pvertex3_array      *vertex_data      /* array of vertex data           */
);

/* QUADRILATERAL MESH WITH DATA */
extern void pquad_mesh_data (
  Pint                      colr_type,        /* colour type                    */
  const Pfacet_array        *facet_data,      /* array of facet data            */
  const Pedge_flag_array    *edge_flags,      /* array of edge flags            */
  const Pvertex_array       *vertex_data      /* array of vertex data           */
);

/* NON-UNIFORM B-SPLINE CURVE 3 */
extern void pnon_uniform_b_spline_curve3 (
  const Pcurve_geom_spline  *curve_spline   /* curve geometry spline          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR */
extern void pnon_uniform_b_spline_curve3_colr (
  const Pcurve_geom_spline  *curve_spline  /* curve geometry spline          */
  const Pcurve_colr_spline  *colr_spline   /* curve colour spline            */
);

/* NON-UNIFORM B-SPLINE SURFACE 3 */
extern void pnon_uniform_b_spline_surf3 (
  const Psurf_geom_spline      *surf_spline,/*surface geometry spline        */
  const Ptrim_curve_loop_list  *trim_loops /* list of trimming curve loops   */
);

/* NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA */
extern void pnon_uniform_b_spline_surf3_data (
  const Psurf_geom_spline       *surf_spline,/*surface geometry spline      */
  const Ptrim_curve_loop_list   *trim_loops,/*list of trimming curve loops  */
  const Psurf_colr_spline       *colr_spline,/*colour spline                */
  const Psurf_data_spline_list  *data_splines/*list of data splines         */
);

/* SET DATA MAPPING INDEX */
extern void pset_data_map_ind (
  Pint                index                /* data mapping index           */
);

/* SET REFLECTANCE INDEX */
extern void pset_refl_ind (
  Pint                index                /* reflectance index            */
);

/* SET BACK INTERIOR INDEX */
extern void pset_back_int_ind (
  Pint                index                /* back interior index          */
);

/* SET BACK DATA MAPPING INDEX */
extern void pset_back_data_map_ind (
  Pint                index                /* back data mapping index      */
);

/* SET BACK REFLECTANCE INDEX */
extern void pset_back_refl_ind (
  Pint                index                /* back reflectance index       */
);

/* SET PARAMETRIC SURFACE INDEX */
extern void pset_param_surf_ind (
  Pint                index                /* parametric surface index     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564232 329 ■


```
/* SET POLYLINE COLOUR */
extern void pset_line_colr (
  const Pgcolr        *colr                /* polyline colour                */
);

/* SET POLYLINE SHADING METHOD */
extern void pset_line_shad_method (
  Pint                method               /* polyline shading method        */
);

/* SET POLYMARKER COLOUR */
extern void pset_marker_colr (
  const Pgcolr        *colr                /* polymarker colour              */
);

/* SET TEXT COLOUR */
extern void pset_text_colr (
  const Pgcolr        *colr                /* text colour                    */
);

/* SET FACET DISTINGUISHING MODE */
extern void pset_facet_disting_mode (
  Pfacet_disting_mode     mode             /* distinguishing mode            */
);

/* SET FACET CULLING MODE */
extern void pset_facet_cull_mode (
  Pfacet_cull_mode    mode                 /* facet culling mode             */
);

/* SET INTERIOR COLOUR */
extern void pset_int_colr (
  const Pgcolr        *colr                /* interior colour                */
);

/* SET INTERIOR SHADING METHOD */
extern void pset_int_shad_method (
  Pint                method               /* interior shading method        */
);

/* SET DATA MAPPING METHOD */
extern void pset_data_map_method (
  const Pdata_map_rep     *method          /* data mapping method            */
);

/* SET REFLECTANCE PROPERTIES */
extern void pset_refl_props (
  const Prefl_prop    *refl_prop           /* reflectance property data      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET REFLECTANCE MODEL */
extern void pset_refl_model (
  Pint                model                 /* reflectance model              */
);

/* SET BACK INTERIOR STYLE */
extern void pset_back_int_style (
  Pint_style          style                 /* back interior style            */
);

/* SET BACK INTERIOR STYLE INDEX */
extern void pset_back_int_style_ind (
  Pint                index                 /* back interior style index      */
);

/* SET BACK INTERIOR COLOUR */
extern void pset_back_int_colr (
  const Pgcolr        *colr                 /* back interior colour           */
);

/* SET BACK INTERIOR SHADING METHOD */
extern void pset_back_int_shad_method (
  Pint                method                /* interior shading method        */
);

/* SET BACK DATA MAPPING METHOD */
extern void pset_back_data_map_method (
  const Pdata_map_rep     *method           /* data mapping method            */
);

/* SET BACK REFLECTANCE PROPERTIES */
extern void pset_back_refl_props (
  const Prefl_prop        *refl_data        /* reflectance property data      */
);

/* SET BACK REFLECTANCE MODEL */
extern void pset_back_refl_model (
  Pint                model                 /* reflectance model              */
);

/* SET LIGHT SOURCE STATE */
extern void pset_light_source_st (
  const Pint_list     *act_list,            /* activation list                */
  const Pint_list     *deact_list           /* deactivation list              */
);

/* SET EDGE COLOUR */
extern void pset_edge_colr (
  const Pgcolr        *colr                 /* edge colour                    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564234 1T1 ■


```
/* SET CURVE APPROXIMATION CRITERIA */
extern void pset_curve_approx_crit (
  const Pcurve_approx_crit    *data          /* curve approximation criteria    */
                                             /* data record                     */
);

/* SET SURFACE APPROXIMATION CRITERIA */
extern void pset_surf_approx_crit (
  const Psurf_approx_crit     *data          /* surface approximation criteria */
                                             /* data record                     */
);

/* SET PARAMETRIC SURFACE CHARACTERISTICS */
extern void pset_param_surf_chars (
  const Pparam_surf_chars     *data          /* parametric surface              */
                                             /* characteristics data record     */
);

/* SET RENDERING COLOUR MODEL */
extern void pset_rend_colr_model (
  Pint                 model                 /* rendering colour model          */
);

/* SET DEPTH CUE INDEX */
extern void pset_depth_cue_ind (
  Pint                 index                 /* depth cue index                 */
);

/* SET COLOUR MAPPING INDEX */
extern void pset_colr_map_ind (
  Pint                 index                 /* colour mapping index            */
);

/* SET POLYLINE REPRESENTATION PLUS */
extern void pset_line_rep_plus (
  Pint                        ws_id,         /* workstation identifier          */
  Pint                        index,         /* polyline bundle index           */
  const Pline_bundle_plus     *rep           /* polyline representation         */
);

/* SET POLYMARKER REPRESENTATION PLUS */
extern void pset_marker_rep_plus (
  Pint                        ws_id,         /* workstation identifier          */
  Pint                        index,         /* polymarker bundle index         */
  const Pmarker_bundle_plus   *rep           /* polymarker representation       */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564235 038 ■


```
/* SET TEXT REPRESENTATION PLUS */
extern void pset_text_rep_plus (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* text bundle index              */
  const Ptext_bundle_plus     *rep          /* text representation            */
);

/* SET INTERIOR REPRESENTATION PLUS */
extern void pset_int_rep_plus (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* interior bundle index          */
  const Pint_bundle_plus      *rep          /* interior representation        */
);

/* SET EDGE REPRESENTATION PLUS */
extern void pset_edge_rep_plus (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* edge bundle index              */
  const Pedge_bundle_plus     *rep          /* edge representation            */
);

/* SET DATA MAPPING REPRESENTATION */
extern void pset_data_map_rep (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* data mapping index             */
  const Pdata_map_rep         *rep          /* data mapping representation    */
);

/* SET REFLECTANCE REPRESENTATION */
extern void pset_refl_rep (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* reflectance index              */
  const Prefl_rep             *rep          /* reflectance representation     */
);

/* SET PARAMETRIC SURFACE REPRESENTATION */
extern void pset_param_surf_rep (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* parametric surface rep. index  */
  const Pparam_surf_rep       *rep          /* parametric surface             */
                                            /* representation                 */
);

/* SET PATTERN REPRESENTATION PLUS */
extern void pset_pat_rep_plus (
  Pint                        ws_id,        /* workstation identifier         */
  Pint                        index,        /* pattern index                  */
  const Ppat_rep_plus         *rep          /* pattern representation plus    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET LIGHT SOURCE REPRESENTATION */
extern void pset_light_source_rep (
  Pint                      ws_id,        /* workstation identifier          */
  Pint                      index,        /* light source index              */
  const Plight_source_rep   *rep          /* light source representation     */
);

/* SET DEPTH CUE REPRESENTATION */
extern void pset_depth_cue_rep (
  Pint                      ws_id,        /* workstation identifier          */
  Pint                      index,        /* depth cue index                 */
  const Pdepth_cue_rep      *rep          /* depth cue representation        */
);

/* SET COLOUR MAPPING REPRESENTATION */
extern void pset_colr_map_rep (
  Pint                      ws_id,        /* workstation identifier          */
  Pint                      index,        /* colour mapping index            */
  const Pcolr_map_rep       *rep          /* colour mapping representation   */
);

/* INQUIRE POLYLINE REPRESENTATION PLUS */
extern void pinq_line_rep_plus (
  Pint              ws_id,                /* workstation identifier          */
  Pint              index,                /* polyline index                  */
  Pinq_type         type,                 /* type of returned values         */
  Pint              *err_ind,             /* OUT error indicator             */
  Pline_bundle_plus *rep                  /* OUT polyline representation     */
);

/* INQUIRE POLYMARKER REPRESENTATION PLUS */
extern void pinq_marker_rep_plus (
  Pint                  ws_id,            /* workstation identifier          */
  Pint                  index,            /* polymarker index                */
  Pinq_type             type,             /* type of returned values         */
  Pint                  *err_ind,         /* OUT error indicator             */
  Pmarker_bundle_plus   *rep              /* OUT polymarker representation   */
);

/* INQUIRE TEXT REPRESENTATION PLUS */
extern void pinq_text_rep_plus (
  Pint              ws_id,                /* workstation identifier          */
  Pint              index,                /* text index                      */
  Pinq_type         type,                 /* type of returned values         */
  Pint              *err_ind,             /* OUT error indicator             */
  Ptext_bundle_plus *rep                  /* OUT text representation         */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE INTERIOR REPRESENTATION PLUS */
extern void pinq_int_rep_plus (
  Pint               ws_id,              /* workstation identifier         */
  Pint               index,              /* interior index                 */
  Pinq_type          type,               /* type of returned values        */
  Pint               *err_ind,           /* OUT error indicator            */
  Pint_bundle_plus   *rep                /* OUT interior representation    */
);

/* INQUIRE EDGE REPRESENTATION PLUS */
extern void pinq_edge_rep_plus (
  Pint               ws_id,              /* workstation identifier         */
  Pint               index,              /* edge index                     */
  Pinq_type          type,               /* type of returned values        */
  Pint               *err_ind,           /* OUT error indicator            */
  Pedge_bundle_plus  *rep                /* OUT edge representation        */
);

/* INQUIRE LIST OF DATA MAPPING INDICES */
extern void pinq_list_data_map_inds (
  Pint               ws_id,              /* workstation identifier         */
  Pint               num_elems_appl_list, /* # elems in appl list          */
  Pint               start_ind,          /* starting index                 */
  Pint               *err_ind,           /* OUT error indicator            */
  Pint_list          *indices,           /* OUT list of defined data       */
                                         /* mapping indices                */
  Pint               *num_elems_impl_list /* OUT # elems in impl list      */
);

/* INQUIRE DATA MAPPING REPRESENTATION */
extern void pinq_data_map_rep (
  Pint               ws_id,              /* workstation identifier         */
  Pint               index,              /* data mapping index             */
  Pinq_type          type,               /* type of returned values        */
  Pstore             store,              /* handle to Store object         */
  Pint               *err_ind,           /* OUT error indicator            */
  Pdata_map_rep      **rep               /* OUT data mapping representation*/
);

/* INQUIRE LIST OF REFLECTANCE INDICES */
extern void pinq_list_refl_inds (
  Pint               ws_id,              /* workstation identifier         */
  Pint               num_elems_appl_list, /* # elems in appl list          */
  Pint               start_ind,          /* starting index                 */
  Pint               *err_ind,           /* OUT error indicator            */
  Pint_list          *indices,           /* OUT list of defined reflectance*/
                                         /* indices                        */
  Pint               *num_elems_impl_list /* OUT # elems in impl list      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE REFLECTANCE REPRESENTATION */
extern void pinq_refl_rep (
  Pint              ws_id,              /* workstation identifier         */
  Pint              index,              /* reflectance index              */
  Pinq_type         type,               /* type of returned values        */
  Pstore            store,              /* handle to Store object         */
  Pint              *err_ind,           /* OUT error indicator            */
  Prefl_rep         **rep               /* OUT reflectance representation */
);

/* INQUIRE LIST OF PARAMETRIC SURFACE INDICES */
extern void pinq_list_param_surf_inds (
  Pint              ws_id,              /* workstation identifier         */
  Pint              num_elems_appl_list, /* # elems in appl list          */
  Pint              start_ind,          /* starting index                 */
  Pint              *err_ind,           /* OUT error indicator            */
  Pint_list         *indices,           /* OUT list of defined parametric */
                                        /* surface indices                */
  Pint              *num_elems_impl_list /* OUT # elems in impl list      */
);

/* INQUIRE PARAMETRIC SURFACE REPRESENTATION */
extern void pinq_param_surf_rep (
  Pint              ws_id,              /* workstation identifier         */
  Pint              index,              /* parametric surface bundle index*/
  Pinq_type         type,               /* type of returned values        */
  Pstore            store,              /* handle to Store object         */
  Pint              *err_ind,           /* OUT error indicator            */
  Pparam_surf_rep   **rep               /* OUT parametric surface         */
                                        /* representation                 */
);

/* INQUIRE PATTERN REPRESENTATION PLUS */
extern void pinq_pat_rep_plus (
  Pint              ws_id,              /* workstation identifier         */
  Pint              index,              /* pattern index                  */
  Pinq_type         type,               /* type of returned values        */
  Pstore            store,              /* handle to Store object         */
  Pint              *err_ind,           /* OUT error indicator            */
  Ppat_rep_plus     **rep               /* OUT pattern representation plus*/
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564239 783 ■


```
/* INQUIRE LIST OF LIGHT SOURCE INDICES */
extern void pinq_list_light_source_inds (
  Pint                ws_id,                /* workstation identifier         */
  Pint                num_elems_appl_list,  /* # elems in appl list           */
  Pint                start_ind,            /* starting index                 */
  Pint                *err_ind,             /* OUT error indicator            */
  Pint_list           *indices,             /* OUT list of defined light      */
                                            /* source indices                 */
  Pint                *num_elems_impl_list  /* OUT # elems in impl list       */
);

/* INQUIRE LIGHT SOURCE REPRESENTATION */
extern void pinq_light_source_rep (
  Pint                ws_id,                /* workstation identifier         */
  Pint                index,                /* light source index             */
  Pinq_type           type,                 /* type of returned values        */
  Pint                *err_ind,             /* OUT error indicator            */
  Plight_source_rep   *rep                  /* OUT light source representation*/
);

/* INQUIRE LIST OF DEPTH CUE INDICES */
extern void pinq_list_depth_cue_inds (
  Pint                ws_id,                /* workstation identifier         */
  Pint                num_elems_appl_list,  /* # elems in appl list           */
  Pint                start_ind,            /* starting index                 */
  Pint                *err_ind,             /* OUT error indicator            */
  Pint_list           *indices,             /* OUT list of defined depth cue  */
                                            /* indices                        */
  Pint                *num_elems_impl_list  /* OUT # elems in impl list       */
);

/* INQUIRE DEPTH CUE REPRESENTATION */
extern void pinq_depth_cue_rep (
  Pint                ws_id,                /* workstation identifier         */
  Pint                index,                /* depth cue index                */
  Pinq_type           type,                 /* type of returned values        */
  Pint                *err_ind,             /* OUT error indicator            */
  Pdepth_cue_rep      *rep                  /* OUT depth cue representation   */
);

/* INQUIRE COLOUR MAPPING STATE */
extern void pinq_colr_map_st (
  Pint                ws_id,                /* workstation identifier         */
  Pint                method,               /* colour mapping method          */
  Pstore              store,                /* handle to Store object         */
  Pint                *err_ind,             /* OUT error indicator            */
  Pcolr_map_st        **data                /* OUT data record                */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564240 4T5 ■


```
/* INQUIRE LIST OF COLOUR MAPPING INDICES */
extern void pinq_list_colr_map_inds (
  Pint               ws_id,               /* workstation identifier         */
  Pint               num_elems_appl_list, /* # elems in appl list           */
  Pint               start_ind,           /* starting index                 */
  Pint               *err_ind,            /* OUT error indicator            */
  Pint_list          *indices,            /* OUT list of defined colour     */
                                          /* mapping indices                */
  Pint               *num_elems_impl_list /* OUT # elems in impl list       */
);

/* INQUIRE COLOUR MAPPING REPRESENTATION */
extern void pinq_colr_map_rep (
  Pint               ws_id,               /* workstation identifier         */
  Pint               index,               /* colour mapping index           */
  Pinq_type          type,                /* type of returned values        */
  Pstore             store,               /* handle to Store object         */
  Pint               *err_ind,            /* OUT error indicator            */
  Pcolr_map_rep      **rep                /* OUT colour mapping             */
                                          /* representation                 */
);

/* INQUIRE DIRECT COLOUR MODEL FACILITIES */
extern void pinq_direct_colr_model_facs (
  Pint           ws_type,                 /* workstation type               */
  Pint           num_elems_appl_list,     /* # elems in appl list           */
  Pint           start_ind,               /* starting index                 */
  Pint           *err_ind,                /* OUT error indicator            */
  Pint_list      *models,                 /* OUT list of directly           */
                                          /* specifiable colour models      */
  Pint           *num_elems_impl_list     /* OUT # elems in impl list       */
);

/* INQUIRE RENDERING COLOUR MODEL FACILITIES */
extern void pinq_rend_colr_model_facs (
  Pint           ws_type,                 /* workstation type               */
  Pint           num_elems_appl_list,     /* # elems in appl list           */
  Pint           start_ind,               /* starting index                 */
  Pint           *err_ind,                /* OUT error indicator            */
  Pint_list      *models,                 /* OUT list of available rendering*/
                                          /* colour models                  */
  Pint           *num_elems_impl_list     /* OUT # elems in impl list       */
);

/* INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS */
extern void pinq_dyns_ws_attrs_plus (
  Pint                   ws_type,         /* workstation type               */
  Pint                   *err_ind,        /* OUT error indicator            */
  Pdyn_ws_attrs_plus     *attr            /* OUT dynamics of workstation    */
                                          /* attributes plus                */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE POLYLINE FACILITIES PLUS */
extern void pinq_line_facs_plus (
  Pint                ws_type,            /* workstation type              */
  Pstore              store,              /* handle to Store object        */
  Pint                *err_ind,           /* OUT error indicator           */
  Pline_facs_plus     **facs              /* OUT list of polyline          */
                                          /* facilities plus               */
);

/* INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS */
extern void pinq_pred_line_rep_plus (
  Pint                    ws_type,        /* workstation type              */
  Pint                    index,          /* predefined polyline index     */
  Pint                    *err_ind,       /* OUT error indicator           */
  Pline_bundle_plus       *rep            /* OUT predefined polyline       */
                                          /* representation                */
);

/* INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS */
extern void pinq_pred_marker_rep_plus (
  Pint                    ws_type,        /* workstation type              */
  Pint                    index,          /* predefined polymarker index   */
  Pint                    *err_ind,       /* OUT error indicator           */
  Pmarker_bundle_plus     *rep            /* OUT predefined polymarker     */
                                          /* representation                */
);

/* INQUIRE PREDEFINED TEXT REPRESENTATION PLUS */
extern void pinq_pred_text_rep_plus (
  Pint                ws_type,            /* workstation type              */
  Pint                index,              /* predefined text index         */
  Pint                *err_ind,           /* OUT error indicator           */
  Ptext_bundle_plus   *rep                /* OUT predefined text           */
                                          /* representation                */
);

/* INQUIRE INTERIOR FACILITIES PLUS */
extern void pinq_int_facs_plus (
  Pint                ws_type,            /* workstation type              */
  Pstore              store,              /* handle to Store object        */
  Pint                *err_ind,           /* OUT error indicator           */
  Pint_facs_plus      **facs              /* OUT list of interior          */
                                          /* facilities plus               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS */
extern void pinq_pred_int_rep_plus (
  Pint                ws_type,              /* workstation type              */
  Pint                index,                /* predefined interior index     */
  Pint                *err_ind,             /* OUT error indicator           */
  Pint_bundle_plus    *rep                  /* OUT predefined interior       */
                                            /* representation                */
);

/* INQUIRE PREDEFINED EDGE REPRESENTATION PLUS */
extern void pinq_pred_edge_rep_plus (
  Pint                ws_type,              /* workstation type              */
  Pint                index,                /* predefined edge index         */
  Pint                *err_ind,             /* OUT error indicator           */
  Pedge_bundle_plus   *rep                  /* OUT predefined edge           */
                                            /* representation                */
);

/* INQUIRE DATA MAPPING FACILITIES */
extern void pinq_data_map_facs (
  Pint           ws_type,                   /* workstation type              */
  Pint           num_elems_appl_list,       /* # elems in appl list          */
  Pint           start_ind,                 /* starting index                */
  Pint           *err_ind,                  /* OUT error indicator           */
  Pint_list      *methods,                  /* OUT list of available data    */
                                            /* mapping methods               */
  Pint           *num_inds,                 /* OUT number of predefined data */
                                            /* mapping indices               */
  Pint           *num_elems_impl_list       /* OUT # elems in impl list      */
);

/* INQUIRE PREDEFINED DATA MAPPING REPRESENTATION */
extern void pinq_pred_data_map_rep (
  Pint                ws_type,              /* workstation type              */
  Pint                index,                /* predefined data mapping index */
  Pstore              store,                /* handle to Store object        */
  Pint                *err_ind,             /* OUT error indicator           */
  Pdata_map_rep       **rep                 /* OUT data mapping representation*/
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE REFLECTANCE FACILITIES */
extern void pinq_refl_facs (
  Pint          ws_type,                /* workstation type              */
  Pstore        store,                  /* handle to Store object        */
  Pint          *err_ind,               /* OUT error indicator           */
  Pint_list     **models,               /* OUT list of available         */
                                        /* reflectance models            */
  Pint_list     **refl_props,           /* OUT list of available         */
                                        /* reflectance properties        */
  Pint          *refl_indices           /* OUT number of predefined      */
                                        /* reflectance indices           */
);

/* INQUIRE PREDEFINED REFLECTANCE REPRESENTATION */
extern void pinq_pred_refl_rep (
  Pint          ws_type,                /* workstation type              */
  Pint          index,                  /* predefined reflectance index  */
  Pstore        store,                  /* handle to Store object        */
  Pint          *err_ind,               /* OUT error indicator           */
  Prefl_rep     **rep                   /* OUT predefined reflectance    */
                                        /* representation                */
);

/* INQUIRE CURVE FACILITIES */
extern void pinq_curve_facs (
  Pint          ws_type,                /* workstation type              */
  Pint          num_elems_appl_list,    /* # elems in appl list          */
  Pint          start_ind,              /* starting index                */
  Pint          *err_ind,               /* OUT error indicator           */
  Pint          *max_order,             /* OUT maximum non-uniform       */
                                        /* b-spline                      */
                                        /* curve order supported         */
  Pint_list     *types,                 /* OUT list of available curve   */
                                        /* approximation criteria types  */
  Pint          *num_elems_impl_list    /* OUT # elems in impl list      */
);

/* INQUIRE PARAMETRIC SURFACE FACILITIES */
extern void pinq_param_surf_facs (
  Pint          ws_type,                /* workstation type              */
  Pint          num_elems_appl_list,    /* # elems in appl list          */
  Pint          start_ind,              /* starting index                */
  Pint          *err_ind,               /* OUT error indicator           */
  Pint_list     *types,                 /* OUT list of available         */
                                        /* parametric surface            */
                                        /* characteristic types          */
  Pint          *num_elems_impl_list,   /* OUT # elems in impl list      */
  Pint          *num_indices            /* OUT number of predefined      */
                                        /* parametric surface indices    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE B-SPLINE SURFACE FACILITIES */
extern void pinq_b_spline_surf_facs (
  Pint           ws_type,                /* workstation type                */
  Pint           num_elems_appl_list,    /* # elems in appl list            */
  Pint           start_ind,              /* starting index                  */
  Pint           *err_ind,               /* OUT error indicator             */
  Pint           *max_order,             /* OUT maximum non-uniform         */
                                         /* b-spline surface order          */
                                         /* supported                       */
  Pint_list      *types,                 /* OUT list of available surface   */
                                         /* approximation criteria types    */
  Pint           *num_elems_impl_list    /* OUT # elems in impl list        */
);

/* INQUIRE TRIMMING CURVE FACILITIES */
extern void pinq_trim_curve_facs (
  Pint           ws_type,                /* workstation type                */
  Pint           num_elems_appl_list,    /* # elems in appl list            */
  Pint           start_ind,              /* starting index                  */
  Pint           *err_ind,               /* OUT error indicator             */
  Pint           *max_order,             /* OUT maximum trimming curve      */
                                         /* order supported                 */
  Pint_list      *types,                 /* OUT list of available trimming  */
                                         /* curve approximation criteria    */
                                         /* types                           */
  Pint           *num_elems_impl_list    /* OUT # elems in impl list        */
);

/* INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION */
extern void pinq_pred_param_surf_rep (
  Pint              ws_type,             /* workstation type                */
  Pint              index,               /* predefined parametric surface   */
                                         /* index                           */
  Pstore            store,               /* handle to Store object          */
  Pint              *err_ind,            /* OUT error indicator             */
  Pparam_surf_rep   **rep                /* OUT parametric surface          */
                                         /* representation                  */
);

/* INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS */
extern void pinq_pred_pat_rep_plus (
  Pint              ws_type,             /* workstation type                */
  Pint              index,               /* predefined pattern index        */
  Pstore            store,               /* handle to Store object          */
  Pint              *err_ind,            /* OUT error indicator             */
  Ppat_rep_plus     **rep                /* OUT pattern representation plus*/
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564245 T87 ■


```
/* INQUIRE LIGHT SOURCE FACILITIES */
extern void pinq_light_source_facs (
  Pint           ws_type,                 /* workstation type                */
  Pint           num_elems_appl_list,     /* # elems in appl list            */
  Pint           start_ind,               /* starting index                  */
  Pint           *err_ind,                /* OUT error indicator             */
  Pint_list      *types,                  /* OUT list of available light     */
                                          /* source types                    */
  Pint           *num_elems_impl_list,    /* OUT # elems in impl list        */
  Pint           *max_lights,             /* OUT maximum number of           */
                                          /* simultaneously active           */
                                          /* non-ambient lights              */
  Pint           *num_indices             /* OUT number of predefined light  */
                                          /* source indices                  */
);

/* INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION */
extern void pinq_pred_light_source_rep (
  Pint                    ws_type,        /* workstation type                */
  Pint                    index,          /* predefined light source index   */
  Pint                    *err_ind,       /* OUT error indicator             */
  Plight_source_rep       *rep            /* OUT light source representation*/
);

/* INQUIRE DEPTH CUE FACILITIES */
extern void pinq_depth_cue_facs (
  Pint                  ws_type,          /* workstation type                */
  Pint                  num_elems_appl_list,/* # elems in appl list          */
  Pint                  start_ind,        /* starting index                  */
  Pint                  *err_ind,         /* OUT error indicator             */
  Pint                  *num_indices      /* OUT number of predefined depth  */
                                          /* cue indices                     */
  Pdepth_cue_mode_list  *modes,           /* OUT list of available depth     */
                                          /* cue modes                       */
  Pint                  *num_elems_impl_list/* OUT # elems impl list         */
);

/* INQUIRE PREDEFINED DEPTH CUE REPRESENTATION */
extern void pinq_pred_depth_cue_rep (
  Pint                ws_type,            /* workstation type                */
  Pint                index,              /* predefined depth cue index      */
  Pint                *err_ind,           /* OUT error indicator             */
  Pdepth_cue_rep      *rep                /* OUT depth cue representation    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564246 913 ■


```
/* INQUIRE COLOUR MAPPING FACILITIES */
extern void pinq_colr_map_facs (
  Pint           ws_type,                /* workstation type                */
  Pint           num_elems_appl_list,    /* # elems in appl list            */
  Pint           start_ind,              /* starting index                  */
  Pint           *err_ind,               /* OUT error indicator             */
  Pint_list      *methods,               /* OUT list of available colour    */
                                         /* mapping methods                 */
  Pint           *num_elems_impl_list    /* OUT # elems in impl list        */
  Pint           *num_indices            /* OUT number of predefined colour*/
                                         /* mapping indices                 */
);

/* INQUIRE COLOUR MAPPING METHOD FACILITIES */
extern void pinq_colr_map_method_facs (
  Pint                   ws_type,        /* workstation type                */
  Pint                   method,         /* colour mapping methods          */
  Pstore                 store,          /* handle to Store object          */
  Pint                   *err_ind,       /* OUT error indicator             */
  Pcolr_map_method       **data          /* OUT data record                 */
);

/* INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION */
extern void pinq_pred_colr_map_rep (
  Pint               ws_type,            /* workstation type                */
  Pint               index,              /* predefined colour mapping index*/
  Pstore             store,              /* handle to Store object          */
  Pint               *err_ind,           /* OUT error indicator             */
  Pcolr_map_rep      **rep               /* OUT colour mapping              */
                                         /* representation                  */
);

/* INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS */
extern void pinq_ws_st_table_length_plus (
  Pint                   ws_type,        /* workstation type                */
  Pint                   *err_ind,       /* OUT error indicator             */
  Pws_st_tables_plus     *lengths        /* OUT lengths of workstation      */
                                         /* tables                          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564247 85T ■

Annex B

(infomative)

Example programs

B.1 star

No changes.

B.2 iron

No changes.

B.3 dyna_star

No changes.

B.4 show_line

No changes.

B.5 xform_pline

No changes.

B.6 surface_trimmed

Clause B.6, add the following PHIGS PLUS example program:

```
/*
 * Program surface_trimmed
 *
 * Copyright notice:
 *   Copyright 1992 O'Reilly and Associates, Inc.  Permission to use, copy,
 *   and modify this program is hereby granted, as long as this copyright
 *   notice appears in each copy of the program source code.
 *
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
 * Description:
 *   This example creates a trimmed non-rational B-spline surface of order 3
 *   in the u-direction and order 4 in the v-direction.
 */

#include <stdio.h>
#include "phigs.h"

#define MODEL (Pint) 1

/* Constants */

const Pint       display_ws_id = 1;   /* id for workstation to display graphics */

/* these two constants should be set to implementation specific values */
const char       *display_conn_id = (void*)""; /* connection id */
const Pint       display_ws_type = 1; /* workstation type */

static void set_view(Pint ws_id); /* defined below */

/* The surface knots and control points. */
static Pfloat uknots[] = { 0, 0, 0, 1, 1, 1 };
static Pfloat vknots[] = { 0, 0, 0, 0, 1, 1, 1, 1 };
static Ppoint3 surf_points[] = {
    {0.0, 0.5, 0.0}, {0.0, 0.6, 0.5}, {0.0, 0.5, 1.0},
    {0.3, 0.5, 0.0}, {0.3, 1.0, 0.5}, {0.3, 0.5, 1.0},
    {0.6, 0.4, 0.0}, {0.6, 0.5, 0.5}, {0.6, 0.4, 1.0},
    {0.9, 0.5, 0.0}, {0.9, 0.3, 0.5}, {0.9, 0.2, 1.0}};

/* Trimming Curves, from inner-most to outer-most. */
static Pfloat t0_knots[] = {0,0,0,1,1,2,2,3,3,4,4,4 };
static Ppoint t0_pts[] = {
    {0.5,0.32},{0.68,0.32},{0.68,0.5},{0.68,0.68},{0.5,0.68},
    {0.32,0.68},{0.32,0.5},{0.32,0.32},{0.5,0.32}};

static Pfloat t1_knots[] = {0,0,0,1,1,2,2,3,3,4,4,4 };
static Ppoint t1_pts[] = {
    {0.5,0.25},{0.75,0.25},{0.75,0.5},{0.75,0.75},{0.5,0.75},
    {0.25,0.75},{0.25,0.5},{0.25,0.25},{0.5,0.25}};

static Pfloat t2_knots[] = { 0, 0, 1, 2, 3, 4, 4 };
static Ppoint t2_pts[] =
    {{0.0, 0.0}, {1.0, 0.0}, {1.0, 1.0}, {0.0, 1.0}, {0.0, 0.0}};

main()
{
    Psurf_geom_spline        surf_spline;
    Ptrim_curve_loop_list    trim_loops;
    Ptrim_curve_loop         trim_loop[3];
    Ptrim_curve              trims[3];
    Pparam_surf_chars        param_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
Psurf_approx_crit    approx_data;

popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);
Popen_ws(display_ws_id, display_conn_id, display_ws_type);

/* Assign the surface parameters. */
surf_spline.u_order = 3;
surf_spline.v_order = 4;
surf_spline.u_knots.num_floats = sizeof(uknots)/sizeof(uknots[0]);
surf_spline.u_knots.floats = uknots;
surf_spline.v_knots.num_floats = sizeof(vknots)/sizeof(vknots[0]);
surf_spline.v_knots.floats = vknots;
surf_spline.rationality = PB_SPLINE_NON_RATIONAL;
surf_spline.ctrl_points.dims.size_x = 3;
surf_spline.ctrl_points.dims.size_y = 4;
surf_spline.ctrl_points.points.point3 = surf_points;

/* Build the trimming loops. */
trim_loops.num_trim_loops = 3;
trim_loops.trim_loops = trim_loop;
trim_loops.trim_loops[0].num_trim_curves = 1;
trim_loops.trim_loops[0].trim_curves = &trims[0];
trim_loops.trim_loops[1].num_trim_curves = 1;
trim_loops.trim_loops[1].trim_curves = &trims[1];
trim_loops.trim_loops[2].num_trim_curves = 1;
trim_loops.trim_loops[2].trim_curves = &trims[2];

trims[0].rationality = PB_SPLINE_NON_RATIONAL;
trims[0].visibility_flag = PEDGE_ON;
trims[0].order = 3;
trims[0].low_limit = 0;
trims[0].high_limit = 4;
trims[0].knots.num_floats = sizeof(t0_knots)/sizeof(t0_knots[0]);
trims[0].knots.floats = t0_knots;
trims[0].ctrl_points.num_points = sizeof(t0_pts)/sizeof(t0_pts[0]);
trims[0].ctrl_points.points.point = t0_pts;
trims[0].approx_crit.type = PSURF_APPROX_CONST_PARAM_SUBD;
trims[0].approx_crit.data.const_param_subd.count = 10;

trims[1].rationality = PB_SPLINE_NON_RATIONAL;
trims[1].visibility_flag = PEDGE_ON;
trims[1].order = 3;
trims[1].low_limit = 0;
trims[1].high_limit = 4;
trims[1].knots.num_floats = sizeof(t1_knots)/sizeof(t1_knots[0]);
trims[1].knots.floats = t1_knots;
trims[1].ctrl_points.num_points = sizeof(t1_pts)/sizeof(t1_pts[0]);
trims[1].ctrl_points.points.point = t1_pts;
trims[1].approx_crit.type = PSURF_APPROX_CONST_PARAM_SUBD;
trims[1].approx_crit.data.const_param_subd.count = 10;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    trims[2].rationality = PB_SPLINE_NON_RATIONAL;
    trims[2].visibility_flag = PEDGE_ON;
    trims[2].order = 2;
    trims[2].low_limit = 0;
    trims[2].high_limit = 4;
    trims[2].knots.num_floats = sizeof(t2_knots)/sizeof(t2_knots[0]);
    trims[2].knots.floats = t2_knots;
    trims[2].ctrl_points.num_points = sizeof(t2_pts)/sizeof(t2_pts[0]);
    trims[2].ctrl_points.points.point = t2_pts;
    trims[2].approx_crit.type = PSURF_APPROX_CONST_PARAM_SUBD;
    trims[2].approx_crit.data.const_param_subd.count = 10;

    /* Create the structure. */
    popen_struct( MODEL );
        pset_view_ind( (Pint) 1 );
        pset_int_style( PSTYLE_EMPTY );

        /* Set the parametric surface characteristics. */
        param_data.type = PPARAM_SURF_CHARS_ISOPARAM;
        param_data.data.isoparam.placement = PPLACE_UNIFORM_OVER_SURF;
        param_data.data.isoparam.num_u_curves = 20;
        param_data.data.isoparam.num_v_curves = 20;
        pset_param_surf_chars( &param_data );

        /* Set the surface approximation criteria. */
        approx_data.type = PSURF_APPROX_CONST_PARAM_SUBD;
        approx_data.data.const_param_subd.u_count = 10;
        approx_data.data.const_param_subd.v_count = 10;
        pset_surf_approx_crit( &approx_data );

        /* Create the surface. */
        pset_edge_flag( PEDGE_ON );
        pnon_uniform_b_spline_surf3( &surf_spline, &trim_loops );
    pclose_struct();

    /* Set the view, and post the top-level structure. */
    set_view( display_ws_id );
    ppost_struct( display_ws_id, MODEL, (Pfloat) 1.0 );
    pupd_ws( display_ws_id, PFLAG_PERFORM );

    sleep( 30 );

    pclose_ws( display_ws_id );
    pclose_phigs();
    return 0;
}

void set_view( Pint ws_id )
{
    int     i,j;
    Pint    error_ind;              /* error indicator */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    Pmatrix3     view_map_matrix;      /* from "peval_view_map_matrix3" */
    Pmatrix3     view_ori_matrix;      /* from "Peval_view_ori_matrix" */

    /* --- viewing parameters --- */
    static Pview_map3 view_mapping = {
        {-0.50, 1.00, -0.50, 1.00 }, /* window limits */
        {0.0,1.0,0.0,1.0,0.0,1.0},   /* projection viewport limits */
        PTYPE_PERSPECT,              /* perspective projection */
        {0.0,0.0,10.0},              /* projection reference point */
        0.0,-2.5,2.5                 /* view/back/front plane distance */
    };
    static Ppoint3 view_ref_pt = {0.0,0.0,0.0};
    static Pvec3 view_norm_vec = {0.4,0.3,1.0};
    static Pvec3 view_up_vec = {0.0,1.0,0.0};
    static Plimit3 clip_limits = {0.0,1.0,0.0,1.0,0.0,1.0};
    Pview_rep3 view_rep;

    peval_view_map_matrix3(&view_mapping, &error_ind, view_map_matrix);
    if(error_ind) {
        fprintf(stderr, "Error %d in peval_view_map_matrix3: Terminating.\n",
                error_ind);
        exit (1);
    }

    peval_view_ori_matrix3(&view_ref_pt, &view_norm_vec, &view_up_vec,
                           &error_ind, view_ori_matrix);
    if (error_ind) {
        fprintf(stderr,"Error %d in peval_view_ori_matrix3: Terminating.\n",
                error_ind);
        exit (1);
    }

    for(i=0;i<4;i++)
        for(j=0;j<4;j++)
            view_rep.ori_matrix[i][j] = view_ori_matrix[i][j];
    for(i=0;i<4;i++)
        for(j=0;j<4;j++)
            view_rep.map_matrix[i][j] = view_map_matrix[i][j];
    view_rep.clip_limit = clip_limits;
    view_rep.xy_clip = view_rep.back_clip = view_rep.front_clip = PIND_CLIP;
    pset_view_rep3(ws_id, 1, &view_rep);
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564252 117 ■

Annex C

(infomative)

Macros for short function identifiers

C.1 Short function identifiers

Clause C.1, merge alphabetically by macro name the following PHIGS PLUS macros for short function identifiers with those of PHIGS:

```
#define pcell_array3_plus                pcap3
#define pfill_area_set_data              pfasd
#define pfill_area_set3_data             pfas3d
#define pinq_b_spline_surf_facs          pqbssf
#define pinq_colr_map_facs               pqcmf
#define pinq_colr_map_method_facs        pqcmmf
#define pinq_colr_map_rep                pqcmr
#define pinq_colr_map_st                 pqcms
#define pinq_curve_facs                  pqcvf
#define pinq_data_map_facs               pqdmf
#define pinq_data_map_rep                pqdmr
#define pinq_depth_cue_facs              pqdcf
#define pinq_depth_cue_rep               pqdcr
#define pinq_direct_colr_model_facs      pqdcmf
#define pinq_dyns_ws_attrs_plus          pqdcwp
#define pinq_edge_rep_plus               pqedp
#define pinq_int_facs_plus               pqifp
#define pinq_int_rep_plus                pqip
#define pinq_light_source_facs           pqlsf
#define pinq_light_source_rep            pqlsr
#define pinq_line_facs_plus              pqplfp
#define pinq_line_rep_plus               pqplp
#define pinq_list_colr_map_inds          pqecmi
#define pinq_list_data_map_inds          pqedmi
#define pinq_list_depth_cue_inds         pqedci
#define pinq_list_light_source_inds      pqelsi
#define pinq_list_param_surf_inds        pqepsi
#define pinq_list_refl_inds              pqerfi
#define pinq_marker_rep_plus             pqmp
#define pinq_param_surf_facs             pqpsf
#define pinq_param_surf_rep              pqpsr
#define pinq_pat_rep_plus                pqpap
#define pinq_pred_colr_map_rep           pqpcmr
#define pinq_pred_data_map_rep           pqpdmr
#define pinq_pred_depth_cue_rep          pqpdcr
#define pinq_pred_edge_rep_plus          pqpedp
#define pinq_pred_int_rep_plus           pqpip
#define pinq_pred_light_source_rep       pqplsr
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define pinq_pred_line_rep_plus                 pqpplp
#define pinq_pred_marker_rep_plus               pqppmp
#define pinq_pred_param_surf_rep                pqppsr
#define pinq_pred_pat_rep_plus                  pqppap
#define pinq_pred_refl_rep                      pqprfr
#define pinq_pred_text_rep_plus                 pqptxp
#define pinq_refl_facs                          pqrff
#define pinq_refl_rep                           pqrfr
#define pinq_rend_colr_model_facs               pqrcmf
#define pinq_text_rep_plus                      pqtxp
#define pinq_trim_curve_facs                    pqtcf
#define pinq_ws_st_table_length_plus            pqwslp
#define pnon_uniform_b_spline_curve3            pbsc3
#define pnon_uniform_b_spline_curve3_colr       pbsc3c
#define pnon_uniform_b_spline_surf3             pbss3
#define pnon_uniform_b_spline_surf3_data        pbss3d
#define ppolyline_set3_colr                     ppls3c
#define pquad_mesh_data                         pqmd
#define pquad_mesh3_data                        pqm3d
#define pset_back_data_map_ind                  psbdmi
#define pset_back_data_map_method               psbdmm
#define pset_back_int_colr                      psbic
#define pset_back_int_ind                       psbii
#define pset_back_int_shad_method               psbism
#define pset_back_int_style_ind                 psbisi
#define pset_back_int_style                     psbis
#define pset_back_refl_ind                      psbrfi
#define pset_back_refl_model                    psbrfm
#define pset_back_refl_props                    psbrfp
#define pset_colr_map_ind                       pscmi
#define pset_colr_map_rep                       pscmr
#define pset_curve_approx_criteria              pscac
#define pset_data_map_ind                       psdmi
#define pset_data_map_method                    psdmm
#define pset_data_map_rep                       psdmr
#define pset_depth_cue_ind                      psdci
#define pset_depth_cue_rep                      psdcr
#define pset_edge_colr                          psedc
#define pset_edge_rep_plus                      psedp
#define pset_facet_cull_mode                    psfcm
#define pset_facet_disting_mode                 psfdm
#define pset_int_colr                           psic
#define pset_int_rep_plus                       psip
#define pset_int_shad_method                    psism
#define pset_light_source_rep                   pslsr
#define pset_light_source_st                    pslss
#define pset_line_colr                          psplc
#define pset_line_rep_plus                      psplp
#define pset_line_shad_method                   psplsm
#define pset_marker_colr                        pspmc
#define pset_marker_rep_plus                    pspmp
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564254 T9T ■


```
#define pset_of_fill_area_sets_data          psfasd
#define pset_of_fill_area_sets3_data         psfas3
#define pset_param_surf_chars                pspsc
#define pset_param_surf_ind                  pspsi
#define pset_param_surf_rep                  pspsr
#define pset_pat_rep_plus                    pspap
#define pset_refl_ind                        psrfi
#define pset_refl_model                      psrfm
#define pset_refl_props                      psrfp
#define pset_refl_rep                        psrfr
#define pset_rend_colr_model                 psrcm
#define pset_surf_approx_criteria            pssac
#define pset_text_colr                       pstxc
#define pset_text_rep_plus                   pstxrp
#define ptri_set_data                        ptsd
#define ptri_set3_data                       pts3d
#define ptri_strip_data                      ptstd
#define ptri_strip3_data                     ptst3d
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564255 926 ■

Annex D

(infomative)

Memory management

D.1 Introduction

No changes.

D.2 Functions that return simple lists

No changes.

D.3 Functions that return complex data structures

No changes.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564256 862 ■

Annex E

(infomative)

Function Lists

E.1 List of functions ordered alphabetically by function name

Clause E.1, merge alphabetically by function name the following list of PHIGS PLUS functions and change the page numbers accordingly:

CELL ARRAY 3 PLUS 63
FILL AREA SET WITH DATA 63
FILL AREA SET 3 WITH DATA 62
INQUIRE B-SPLINE SURFACE FACILITIES 90
INQUIRE COLOUR MAPPING FACILITIES 93
INQUIRE COLOUR MAPPING METHOD FACILITIES 93
INQUIRE COLOUR MAPPING REPRESENTATION 84
INQUIRE COLOUR MAPPING STATE 83
INQUIRE CURVE FACILITIES 90
INQUIRE DATA MAPPING FACILITIES 88
INQUIRE DATA MAPPING REPRESENTATION 80
INQUIRE DEPTH CUE FACILITIES 92
INQUIRE DEPTH CUE REPRESENTATION 83
INQUIRE DIRECT COLOUR MODEL FACILITIES 85
INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES PLUS 85
INQUIRE EDGE REPRESENTATION PLUS 79
INQUIRE INTERIOR FACILITIES PLUS 87
INQUIRE INTERIOR REPRESENTATION PLUS 79
INQUIRE LIGHT SOURCE FACILITIES 92
INQUIRE LIGHT SOURCE REPRESENTATION 82
INQUIRE LIST OF COLOUR MAPPING INDICES 84
INQUIRE LIST OF DATA MAPPING INDICES 80
INQUIRE LIST OF DEPTH CUE INDICES 83
INQUIRE LIST OF LIGHT SOURCE INDICES 82
INQUIRE LIST OF PARAMETRIC SURFACE INDICES 81
INQUIRE LIST OF REFLECTANCE INDICES 80
INQUIRE PARAMETRIC SURFACE FACILITIES 90
INQUIRE PARAMETRIC SURFACE REPRESENTATION 81
INQUIRE PATTERN REPRESENTATION PLUS 82
INQUIRE POLYLINE FACILITIES PLUS 86
INQUIRE POLYLINE REPRESENTATION PLUS 78
INQUIRE POLYMARKER REPRESENTATION PLUS 78
INQUIRE PREDEFINED COLOUR MAPPING REPRESENTATION 94
INQUIRE PREDEFINED DATA MAPPING REPRESENTATION 88
INQUIRE PREDEFINED DEPTH CUE REPRESENTATION 93
INQUIRE PREDEFINED EDGE REPRESENTATION PLUS 88
INQUIRE PREDEFINED INTERIOR REPRESENTATION PLUS 87

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564257 7T9 ■

INQUIRE PREDEFINED LIGHT SOURCE REPRESENTATION 92
INQUIRE PREDEFINED PARAMETRIC SURFACE REPRESENTATION 91
INQUIRE PREDEFINED PATTERN REPRESENTATION PLUS 91
INQUIRE PREDEFINED POLYLINE REPRESENTATION PLUS 86
INQUIRE PREDEFINED POLYMARKER REPRESENTATION PLUS 86
INQUIRE PREDEFINED REFLECTANCE REPRESENTATION 89
INQUIRE PREDEFINED TEXT REPRESENTATION PLUS 87
INQUIRE REFLECTANCE FACILITIES 89
INQUIRE REFLECTANCE REPRESENTATION 81
INQUIRE RENDERING COLOUR MODEL FACILITIES 85
INQUIRE TEXT REPRESENTATION PLUS 79
INQUIRE TRIMMING CURVE FACILITIES 91
INQUIRE WORKSTATION STATE TABLE LENGTHS PLUS 94
NON-UNIFORM B-SPLINE CURVE 3 66
NON-UNIFORM B-SPLINE SURFACE 3 67
NON-UNIFORM B-SPLINE SURFACE 3 WITH DATA 67
NON-UNIFORM B-SPLINE CURVE 3 WITH COLOUR 67
POLYLINE SET 3 WITH COLOUR 62
QUADRILATERAL MESH WITH DATA 66
QUADRILATERAL MESH 3 WITH DATA 66
SET BACK DATA MAPPING INDEX 68
SET BACK DATA MAPPING METHOD 72
SET BACK INTERIOR COLOUR 72
SET BACK INTERIOR INDEX 68
SET BACK INTERIOR SHADING METHOD 72
SET BACK INTERIOR STYLE INDEX 72
SET BACK INTERIOR STYLE 71
SET BACK REFLECTANCE INDEX 69
SET BACK REFLECTANCE MODEL 73
SET BACK REFLECTANCE PROPERTIES 72
SET COLOUR MAPPING INDEX 75
SET COLOUR MAPPING INDEX 75
SET COLOUR MAPPING REPRESENTATION 78
SET CURVE APPROXIMATION CRITERIA 73
SET DATA MAPPING INDEX 68
SET DATA MAPPING METHOD 71
SET DATA MAPPING REPRESENTATION 76
SET DEPTH CUE INDEX 74
SET DEPTH CUE REPRESENTATION 77
SET EDGE COLOUR 73
SET EDGE REPRESENTATION PLUS 76
SET FACET CULLING MODE 70
SET FACET DISTINGUISHING MODE 70
SET INTERIOR COLOUR 70
SET INTERIOR REPRESENTATION PLUS 76
SET INTERIOR SHADING METHOD 71
SET LIGHT SOURCE REPRESENTATION 77
SET LIGHT SOURCE STATE 73
SET OF FILL AREA SETS WITH DATA 64
SET OF FILL AREA SETS 3 WITH DATA 63
SET PARAMETRIC SURFACE CHARACTERISTICS 74

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

9593 PT*4 91 ■ 4851903 0564258 635 ■

SET PARAMETRIC SURFACE INDEX 69
SET PARAMETRIC SURFACE REPRESENTATION 77
SET PATTERN REPRESENTATION PLUS 77
SET POLYLINE COLOUR 69
SET POLYLINE REPRESENTATION PLUS 75
SET POLYLINE SHADING METHOD 69
SET POLYMARKER COLOUR 70
SET POLYMARKER REPRESENTATION PLUS 75
SET REFLECTANCE INDEX 68
SET REFLECTANCE MODEL 71
SET REFLECTANCE PROPERTIES 71
SET REFLECTANCE REPRESENTATION 76
SET RENDERING COLOUR MODEL 74
SET SURFACE APPROXIMATION CRITERIA 74
SET TEXT COLOUR 70
SET TEXT REPRESENTATION PLUS 75
TRIANGLE SET WITH DATA 65
TRIANGLE SET 3 WITH DATA 64
TRIANGLE STRIP WITH DATA 65
TRIANGLE STRIP 3 WITH DATA 65

E.2 List of functions ordered alphabetically by bound name

Clause E.2, merge alphabetically by bound name the following list of PHIGS PLUS functions and change the page numbers accordingly:

pcell_array3_plus (...) 63
pfill_area_set_data (...) 63
pfill_area_set3_data (...) 62
pinq_b_spline_surf_facs (...) 90
pinq_colr_map_facs (...) 93
pinq_colr_map_method_facs (...) 93
pinq_colr_map_rep (...) 84
pinq_colr_map_st (...) 83
pinq_curve_facs (...) 90
pinq_data_map_facs (...) 88
pinq_data_map_rep (...) 80
pinq_depth_cue_facs (...) 92
pinq_depth_cue_rep (...) 83
pinq_direct_colr_model_facs (...) 85
pinq_dyns_ws_attrs_plus (...) 85
pinq_edge_rep_plus (...) 79
pinq_int_facs_plus (...) 87
pinq_int_rep_plus (...) 79
pinq_light_source_facs (...) 92
pinq_light_source_rep (...) 82
pinq_line_facs_plus (...) 86
pinq_line_rep_plus (...) 78
pinq_list_colr_map_inds (...) 84

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564259 571 ■

pinq_list_data_map_inds (...) 80
pinq_list_depth_cue_inds (...) 83
pinq_list_light_source_inds (...) 82
pinq_list_param_surf_inds (...) 81
pinq_list_refl_inds (...) 80
pinq_marker_rep_plus (...) 78
pinq_param_surf_facs (...) 90
pinq_param_surf_rep (...) 81
pinq_pat_rep_plus (...) 82
pinq_pred_colr_map_rep (...) 94
pinq_pred_data_map_rep (...) 88
pinq_pred_depth_cue_rep (...) 93
pinq_pred_edge_rep_plus (...) 88
pinq_pred_int_rep_plus (...) 87
pinq_pred_light_source_rep (...) 92
pinq_pred_line_rep_plus (...) 86
pinq_pred_marker_rep_plus (...) 86
pinq_pred_param_surf_rep (...) 91
pinq_pred_pat_rep_plus (...) 91
pinq_pred_refl_rep (...) 89
pinq_pred_text_rep_plus (...) 87
pinq_refl_facs (...) 89
pinq_refl_rep (...) 81
pinq_rend_colr_model_facs (...) 85
pinq_text_rep_plus (...) 79
pinq_trim_curve_facs (...) 91
pinq_ws_st_table_length_plus (...) 94
pnon_uniform_b_spline_curve3 (...) 66
pnon_uniform_b_spline_curve3_colr (...) 67
pnon_uniform_b_spline_surf3 (...) 67
pnon_uniform_b_spline_surf3_data (...) 67
ppolyline_set3_colr (...) 62
pquad_mesh_data (...) 66
pquad_mesh3_data (...) 66
pset_back_data_map_ind (...) 68
pset_back_data_map_method (...) 72
pset_back_int_colr (...) 72
pset_back_int_ind (...) 68
pset_back_int_shad_method (...) 72
pset_back_int_style_ind (...) 72
pset_back_int_style (...) 71
pset_back_refl_ind (...) 69
pset_back_refl_model (...) 73
pset_back_refl_props (...) 72
pset_colr_map_ind (...) 75
pset_colr_map_rep (...) 78
pset_curve_approx_crit (...) 73
pset_data_map_ind (...) 68
pset_data_map_method (...) 71
pset_data_map_rep (...) 76
pset_depth_cue_ind (...) 74

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

9593 PT*4 91 ■ 4851903 0564260 293 ■

pset_depth_cue_rep (...) 77
pset_edge_colr (...) 73
pset_edge_rep_plus (...) 76
pset_facet_cull_mode (...) 70
pset_facet_disting_mode (...) 70
pset_int_colr (...) 70
pset_int_rep_plus (...) 76
pset_int_shad_method (...) 71
pset_light_source_rep (...) 77
pset_light_source_st (...) 73
pset_line_colr (...) 69
pset_line_rep_plus (...) 75
pset_line_shad_method (...) 69
pset_marker_colr (...) 70
pinq_marker_rep_plus (...) 78
pset_of_fill_area_sets_data (...) 64
pset_of_fill_area_sets3_data (...) 63
pset_param_surf_chars (...) 74
pset_param_surf_ind (...) 69
pset_param_surf_rep (...) 77
pset_pat_rep_plus (...) 77
pset_refl_ind (...) 68
pset_refl_model (...) 71
pset_refl_props (...) 71
pset_refl_rep (...) 76
pset_rend_colr_model (...) 74
pset_surf_approx_crit (...) 74
pset_text_colr (...) 70
pset_text_rep_plus (...) 75
ptri_set_data (...) 65
ptri_set3_data (...) 64
ptri_strip_data (...) 65
ptri_strip3_data (...) 65

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

© ISO/IEC

ICS 35.140.00; 35.060.00

Descriptors: data processing, information interchange, graphic data processing, programming (computers), programming languages, C (programming language).

Price based on 157 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577329 395 ■

INTERNATIONAL STANDARD

ISO/IEC 9593-4

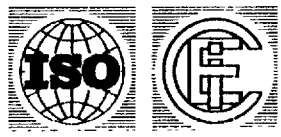
First edition
1991-12-15

Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings —

Part 4:
C

Technologies de l'information — Infographie — Interfaces langage entre un programme d'application et son support graphique —

Partie 4: C



Reference number
ISO/IEC 9593-4:1991(E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577330 007 ■

Contents

1 Scope 1
2 Normative references 2
3 The C language binding of PHIGS 3
3.1 Conformance 3
3.2 Functions versus macros 3
3.3 Character strings 3
3.4 Function identifiers 3
3.5 Registration 3
3.6 Identifiers for graphics items 3
3.7 Return values 4
3.8 Header files 4
3.9 Memory management 5
3.9.1 Inquiry functions which return simple lists 5
3.9.2 Inquiry functions which return complex data structures 5
3.9.3 Meaning of the size of an element 7
3.10 Inquiries returning structure elements 7
3.11 Error handling 7
3.11.1 Application defined error handlers 7
3.11.2 Error codes 7
3.11.3 C specific PHIGS errors 8
3.12 Storage of two-dimensional data 8
3.12.1 Storage of matrices 8
3.12.2 Storage of colour arrays 9
4 Tables 10
4.1 Abbreviation policy for construction of identifiers 10
4.2 Table of abbreviations 10
4.3 Function names 14
4.3.1 List ordered alphabetically by bound name 14
4.3.2 List ordered alphabetically by PHIGS function name 20
5 Type definitions 27
5.1 Mapping of PHIGS data types 27
5.2 Environmental type definitions 27
5.3 Implementation dependent type definitions 28
5.4 Implementation independent type definitions 38
6 Macro definitions 73
6.1 Function identifiers 73
6.2 Error codes 76
6.3 Miscellaneous 83
6.3.1 Linetypes 83
6.3.2 Marker types 83
6.3.3 Annotation styles 83
6.3.4 Colour models 83
6.3.5 Prompt and Echo Types 83
6.3.6 Default parameters of OPEN PHIGS 84
6.3.7 Element enumeration 84

© ISO/IEC 1991
All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Foreword

7 C PHIGS functions 85
7.1 Notational conventions 85
7.2 Control functions 85
7.3 Output primitive functions 87
7.4 Attribute specification functions 89
7.4.1 Bundled attribute selection 89
7.4.2 Individual attribute selection 90
7.4.3 Aspect source flag setting 95
7.4.4 Workstation attribute table definition 95
7.4.5 Workstation filter definition 96
7.4.6 Colour model control 97
7.4.7 HLHSR attributes 97
7.5 Transformation and clipping functions 97
7.5.1 Modelling transformations and clipping 97
7.5.2 View operation 99
7.5.3 Workstation transformation 99
7.5.4 Utility functions to support modelling 100
7.5.5 Utility functions to support viewing 104
7.6 Structure content functions 105
7.7 Structure manipulation functions 107
7.8 Structure display functions 108
7.9 Structure archiving functions 108
7.10 Input functions 111
7.10.1 Pick identifier and filter 111
7.10.2 Initialization of input devices 112
7.10.3 Setting the mode of input devices 115
7.10.4 Request input functions 116
7.10.5 Sample input functions 118
7.10.6 Event input functions 120
7.11 Metafile functions 122
7.12 Inquiry functions 123
7.12.1 Inquiry functions for operating state values 123
7.12.2 Inquiry functions for PHIGS description table 123
7.12.3 Inquiry functions for PHIGS state list 125
7.12.4 Inquiry functions for workstation state list 126
7.12.5 Inquiry functions for workstation description table 137
7.12.6 Inquiry function for structure state list 149
7.12.7 Inquiry functions for structure content 150
7.12.8 Inquiry functions for error state list 153
7.13 Error control 153
7.14 Special interfaces 154
7.15 Binding defined utility functions 154
Annexes
A Data types in compilation order and external functions 155
A.1 Macro definitions 155
A.2 Types in compilation order 166
A.3 External functions 196
B Example Programs 248
B.1 star 248
B.2 iron 251
B.3 dyna_star 258
B.4 show_line 264
B.5 xform_pline 269
C Macros for short function identifiers 277

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

C.1 Short function identifiers 277
D Memory management 284
D.1 Introduction 284
D.2 Functions that return simple lists 284
D.2.1 Operation of `ping_list_line_inds` 285
D.3 Functions that return complex data structures 288
D.3.1 Operation of `pcreate_store` 289
D.3.2 Operation of `pinq_stroke_st` and `pinq_pat_rep` 291
D.3.3 Operation of `pdel_store` 295
E Function Lists 297
E.1 List of functions ordered alphabetically by function name 297
E.2 List of functions ordered alphabetically by bound name 303

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9593-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

ISO/IEC 9593 consists of the following parts, under the general title *Information technology — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings* :

— *Part 1 : FORTRAN 77*
— *Part 3 : ADA*
— *Part 4 : C*

Annexes A, B, C, D, and E of this part of ISO/IEC 9593 are for information only.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Introduction

The Programmer's Hierarchical Interactive Graphics System (PHIGS), the functional description of which is given in ISO/IEC 9592-1, is specified in a language independent manner and needs to be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this part of ISO/IEC 9593 is to define a standard binding for the C computer programming language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577335 699 ■

Information technology – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings –

Part 4:

C

1 Scope

The Programmer's Hierarchical Interactive Graphics System (PHIGS), ISO/IEC 9592-1, specifies a language independent nucleus of a graphics system. For integration into a programming language, PHIGS is embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 9593 specifies such a language dependent layer for the C language.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577336 525 ■

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9593. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9593 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 9592-1 : 1989, *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 1:Functional description.*

ISO/IEC 9899 : 1990, *Programming Languages – C.*

ISO/IEC TR 9973 : 1988, *Information processing – Procedures for Registration of Graphical Items.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577337 461 ■

3 The C language binding of PHIGS

3.1 Conformance

This binding incorporates the rules of conformance defined in the PHIGS Standard (ISO/IEC 9592) for PHIGS implementations, with those additional requirements specifically defined for C language implementations of PHIGS.

The following criteria are established for determining conformance of an implementation to this binding:

In order to conform, an implementation of the C binding of PHIGS shall implement ISO/IEC 9592-1. It shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 9593.

Thus, for example, the syntax of the function names shall be precisely as specified in this part of ISO/IEC 9593 and the parameters shall be of the data types stated in this part of ISO/IEC 9593.

3.2 Functions versus macros

An implementation may substitute macros for functions. However, the macros must be designed so that side-effects work properly.

3.3 Character strings

The C language represents character strings as an array of characters terminated by the null character (*i.e.* '\0'). This means that the null character is not usable as a printable character.

3.4 Function identifiers

The C standard (ISO/IEC 9899) requires that compilers recognize internal identifiers which are distinct in at least 31 characters. That standard also requires that external identifiers (*i.e.* those seen by the linker) be recognized to a minimum of six characters, independent of case.

In order to produce more readable C code, this part of ISO/IEC 9593 binds the PHIGS function names to identifiers which can be longer than six characters but are less than 31 characters. Implementations which must run in environments which honour less than 31 characters in external identifiers must include a set of `#defines` in the header file which equate the long identifiers defined in this part of ISO/IEC 9593 to a set of shorter identifiers which conform to the requirements of the implementation. An example of these shorter names can be found in annex C.

3.5 Registration

ISO/IEC 9592-1 reserves certain value ranges for registration[1] as graphical items. The registered graphical items will be bound to the C programming language (and other programming languages). The binding of registered items will be consistent with the bindings presented in this part of ISO/IEC 9593.

3.6 Identifiers for graphics items

ISO/IEC 9592-1 provides a mechanism to extend its functionality by the use of the graphical items GENERALIZED DRAWING PRIMITIVE, GENERALIZED DRAWING PRIMITIVE 3, GENERALIZED STRUCTURE ELEMENT and ESCAPE. These items are referenced via identifiers formed from the registration number of the item. This binding specifies the format of the identifiers but it does not specify the registration of the identifiers. The identifiers are used as arguments to the functions `pgdp`, `pgdp3`, `pgse`, and `pescape` and returned to the application by the functions `pinq_elem_content` and `pinq_cur_elem_content`.

[1] For the purpose of this International Standard and according to the rules for the designation and operation of registration authorities in the ISO/IEC JTC1 procedures, the ISO and IEC Councils have designated the National Institute of Standards and Technology (National Computer Systems Laboratory), A-266 Technology Building, Gaithersburg, MD 20899, USA, to act as registration authority.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

An implementation may also represent these items as separate functions, but this is not required.

The format of the identifiers for graphical items is

"PFFF_Tn"

where:

FFF is the abbreviation of the PHIGS function name.

T is the type of the item, "R" for registered and "U" for unregistered.

n is the registration number (for unregistered items, the absolute value of the item number shall be used).

The formats for registered GENERALIZED DRAWING PRIMITIVE, GENERALIZED DRAWING PRIMITIVE 3, GENERALIZED STRUCTURE ELEMENT and ESCAPE graphical items are:

```
#define PGDP_Rn                 (n)
#define PGDP3_Rn                (n)
#define PGSE_Rn                 (n)
#define PESCAPE_Rn              (n)
```

For example, the identifier for registered GENERALIZED DRAWING PRIMITIVE item 12 would be:

```
#define PGDP_R12                (12)
```

The formats for unregistered GENERALIZED DRAWING PRIMITIVE, GENERALIZED DRAWING PRIMITIVE 3, GENERALIZED STRUCTURE ELEMENT and ESCAPE graphical items are:

```
#define PGDP_Un                 (-n)
#define PGDP3_Un                (-n)
#define PGSE_Un                 (-n)
#define PESCAPE_Un              (-n)
```

For example, the identifier for unregistered GENERALIZED DRAWING PRIMITIVE item -1 would be:

```
#define PGDP_U1                 (-1)
```

3.7 Return values

All PHIGS C functions return `void`.

3.8 Header files

C provides a mechanism to allow external files to be included in a compilation. The file `phigs.h` shall be included in any application program that intends to use PHIGS via the C binding.

Clause 5 of this binding describes the data types that shall be defined in the file `phigs.h`; clause 6 defines the macros that shall be in the file `phigs.h`. Additional implementation-dependent items may be placed in this file if needed.

The type `size_t` is used in the binding. It is defined in the standard header `<stddef.h>`. Therefore, the file `phigs.h` shall include `<stddef.h>` in order to define `size_t`.

The file `phigs.h` shall also contain external prototype declarations for all PHIGS C functions because they return `void`. For example, the declaration for the function `popen_phigs` is:

```
extern   void   popen_phigs(const char *err_file, size_t mem_units);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

3.9 Memory management

The application shall allocate the memory needed for the data returned by the implementation. In general, the application will allocate a C structure and pass a pointer to that structure to an inquiry routine which will then place information into the structure. However, a number of inquiry functions return variable length data, the length of which is not known *a priori* by the application.

These functions fall into two classes. One class of functions returns a simple, homogeneous list of elements. For example, the function INQUIRE STRUCTURE IDENTIFIERS returns a list of the structure identifiers in use. The other class returns complex, heterogeneous data structures. For example, the function INQUIRE LOCATOR DEVICE STATE returns the device state which includes a locator data record; the data record can contain arbitrarily complex implementation-defined data structures. The binding of these two classes of functions is described in detail below.

Additional binding-specific errors which relate to memory management are described in 3.11.3.

3.9.1 Inquiry functions which return simple lists

Inquiry functions which return a list of elements are bound such that the application can inquire about a portion of the list. This list portion is a subset of the implementation's internal list and is called the application's list. This allows the application to process the implementation's list in a piecewise manner rather than all at once.

The application allocates the memory for its list and passes that list to the implementation. The implementation places the results of the inquiry into the list. In order to support this policy of memory management, three additional parameters have been added to functions which return simple lists:

a) `num_elem_appl_list`: An integer input parameter which is the size of the application's list. The value of `num_elem_appl_list` indicates the number of list elements which will fit into the application's list. A value of 0 is valid and allows the application to determine the size of the implementation's list (which is returned via `num_elem_impl_list`) without having the implementation return any of the elements of its list.

b) `start_ind`: An integer input parameter which is the starting index into the implementation's list. (Index 0 is the first element of both the implementation's and application's list.) `start_ind` indicates the element in the implementation's list that is copied into index 0 of the application's list. Elements are copied sequentially from the implementation's list into the application's list until the application's list is full or there are no more elements in the implementation's list.

   If `start_ind` is out of range, error number 2200 (`PE_START_IND_INVAL`) is returned as the value of the error indicator parameter.

c) `num_elem_impl_list`: An output parameter which is a pointer to an integer. The implementation stores into this parameter the number of elements that are in the implementation's list.

Each function which returns a simple list has an output parameter *list*, which is a pointer to a structure with fields for the number of elements in the list and a pointer to the elements in the list. The type of *list* depends upon the function in which it is used. The implementation places the elements in the memory pointed to by *list->elems* and assigns the number of elements in *list->elems* to *list->num_elems*. If the implementation's list is of zero length, no data is placed in *list->elems* and *list->num_elems* is set to zero.

3.9.2 Inquiry functions which return complex data structures

The data returned by the element content functions and the functions which return input device data records can be complex in structure; they cannot be represented by a simple list of elements. It would be an onerous task for the application to allocate and to prepare data structures for these routines. In order to facilitate the task of using these inquiry functions, the binding defines a new resource, called a *Store*, to manage the memory for these functions.

A *Store* is used by the implementation to manage the memory needed by the functions which return complex data structures. The *Store* resource is opaque to the application. The application does not know the structure of the *Store* or how it is implemented. The *Store* is defined as a `void *`. The binding defines two new functions which create

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

(CREATE STORE, bound as `pcreate_store`) and destroy (DELETE STORE, bound as `pdel_store`) a *Store*.

The semantics of the *Store* resource provide two levels of memory management: The implementation is responsible for managing the memory at a low level because it uses, re-uses, allocates and deallocates memory from the system in order to return information to the application. But the application is ultimately responsible for managing the memory at a high level because it creates and destroys *Stores*.

A *Store* is passed as a parameter to a function returning complex data. Another parameter to this function is a pointer to a pointer to a structure which defines the format of the returned data. The *Store* contains memory for the structure and any additional memory referenced by fields within the structure. The application accesses the returned data through its pointer to the structure; it does not use the *Store* to access the data.

For some functions, a *Store* is used to manage the memory for two or more distinct complex data structures. For example, in the function INQUIRE PICK DEVICE STATE, the *Store* manages the memory for the pick filter, the initial pick path, and the pick data record, all of which are returned to the application.

A *Store* continues to hold the information returned from the function until the *Store* is destroyed by the `pdel_store` function, or until the *Store* is used as an argument to a subsequent function which returns complex data. At that time, the old information is replaced with the new. Thus multiple calls to functions overwrite the contents of a *Store*. A *Store* only contains the results of the last function. An application may create more than one *Store*.

This binding defines two new errors that can occur when using or creating a *Store*; these errors are described in 3.11.3. For most functions using a *Store*, these and other errors are returned via the "error indicator" parameter. However, the functions RETRIEVE PATHS TO ANCESTORS, RETRIEVE PATHS TO DESCENDANTS and ESCAPE do not have an error indicator parameter. For these functions, the error reporting mechanism is used when an error is encountered. For these functions, the implementation shall, in addition to reporting the error, set the pointer to the returned data to NULL when an error occurs.

The definitions for the functions CREATE STORE and DELETE STORE follow:

CREATE STORE (PHOP, *, *, *)

Parameters:

OUT	error indicator	I
OUT	store	STORE

Effect: Creates a Store and returns a handle to it in the parameter *store*. If the Store cannot be created, the *store* parameter is set to NULL and the error indicator is set to one of the following error numbers:

002 Ignoring function, function requires state (PHOP, *, *, *).

2203 Ignoring function, error allocating Store.

Errors: none

DELETE STORE (PHOP, *, *, *)

Parameters:

OUT	error indicator	I
IN/OUT	store	STORE

Effect: Deletes the Store and all internal resources associated with it. The parameter *store* is set to NULL to signify that it is no longer valid. If an error is detected, the error indicator is set to the following error number:

002 Ignoring function, function requires state (PHOP, *, *, *).

Errors: none

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577341 992 ■

3.9.3 Meaning of the size of an element

The functions INQUIRE CURRENT ELEMENT TYPE AND SIZE and INQUIRE ELEMENT TYPE AND SIZE return the size of an element. The size of an element is the size of a buffer, in bytes, that the application would have to allocate in order to contain the element. If the application would not have to allocate any memory, then 0 is returned.

3.10 Inquiries returning structure elements

PHIGS provides the ability for the application to inquire the contents of a structure element (through INQUIRE CURRENT ELEMENT CONTENT and INQUIRE ELEMENT CONTENT). PHIGS also allows the application to read in an archive file that can contain GDPs (both GENERALIZED DRAWING PRIMITIVEs and GENERALIZED DRAWING PRIMITIVE 3's) and GSEs (GENERALIZED STRUCTURE ELEMENTs) which are not supported by the implementation. Thus it is possible for the application to inquire the contents of a structure element that contains a GDP or GSE data record that the implementation does not support.

In order to allow the inquiry of unsupported data records, the binding has introduced a field, called `unsupp`, to the GDP and GSE data record structures. The type of this field is `Pdata` which is a structure containing a field for the size of a block of data and another field for a pointer to the data. The `unsupp` field is used if the implementation does not support a GDP or a GSE.

3.11 Error handling

3.11.1 Application defined error handlers

An application can define the error handling function for the PHIGS implementation via the utility function SET ERROR HANDLER, which is bound as `pset_err_hand`. The definition for SET ERROR HANDLER is:

SET ERROR HANDLER (*, *, *, *)

Parameters:

IN	new error handling function	Function
OUT	old error handling function	Function

Effect: Sets the PHIGS error handling function to *new error handling function* and returns the previous function in *old error handling function.*

Errors: none

Application defined error handling functions accept the same arguments as the standard error handler. It may invoke the standard error logging function `perr_log`.

ISO/IEC 9592-1 defines the initial error handling function to be `perr_hand`; that is, the value of the parameter `old_err_hand` points to `perr_hand` when SET ERROR HANDLER is invoked for the first time.

When the application changes the error handling function, the implementation will invoke the new function when an error is detected. If the application calls the default error handling function `perr_hand`, `perr_hand` will always call the default error logging function `perr_log`; `perr_hand` does not call the error handling function specified in SET ERROR HANDLER.

3.11.2 Error codes

This binding defines, in 6.2, a set of constants for the PHIGS error numbers. Each error constant begins with the characters PE.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

3.11.3 C specific PHIGS errors

This binding defines the following additional errors, beyond the ones described in ISO/IEC 9592-1.

Error	Message
2200	`Ignoring function, start index is out of range` Is issued when the start index is less than zero or larger than the last element in the implementation list.
2201	`Ignoring function, length of application's list is negative` Is issued when the length of the application's list is less than zero.
2202	`Ignoring function, enumeration type out of range` Is issued when a parameter value whose type is an enumeration is out range of the enumeration.
2203	`Ignoring function, error while allocating a Store` Is issued when an error is detected during CREATE STORE.
2204	`Ignoring function, error while allocating memory for a Store` Is issued when a function using a Store is unable to allocate memory for the store.

3.12 Storage of two-dimensional data

3.12.1 Storage of matrices

ISO/IEC 9592-1 represents transformations as 3×3 and 4×4 matrices. This part of ISO/IEC 9593 binds a PHIGS 3×3 matrix to the type `Pmatrix` and a PHIGS 4×4 matrix to the type `Pmatrix3`.

The elements of a PHIGS 3×3 matrix are:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

and these elements are stored such that:

```
m[0][0] = a;    m[0][1] = b;    m[0][2] = c;
m[1][0] = d;    m[1][1] = e;    m[1][2] = f;
m[2][0] = g;    m[2][1] = h;    m[2][2] = i;
```

where `m` is of type `Pmatrix`.

The elements of a PHIGS 4×4 matrix are:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$$

and these elements are stored such that:

```
m[0][0] = a;    m[0][1] = b;    m[0][2] = c;    m[0][3] = d;
m[1][0] = e;    m[1][1] = f;    m[1][2] = g;    m[1][3] = h;
m[2][0] = i;    m[2][1] = j;    m[2][2] = k;    m[2][3] = l;
m[3][0] = m;    m[3][1] = n;    m[3][2] = o;    m[3][3] = p;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

where m is of type Pmatrix3.

For example, in order to store the projection transformation defined by:

$$x \rightarrow x'/w'$$
$$y \rightarrow y'/w'$$
$$z \rightarrow z'/w'$$

in PHIGS 4×4 matrix, the values shall be stored such that:

```
x' = p[0][0]*x + p[0][1]*y + p[0][2]*z + p[0][3];
y' = p[1][0]*x + p[1][1]*y + p[1][2]*z + p[1][3];
z' = p[2][0]*x + p[2][1]*y + p[2][2]*z + p[2][3];
w' = p[3][0]*x + p[3][1]*y + p[3][2]*z + p[3][3];
```

where p is of type Pmatrix3.

3.12.2 Storage of colour arrays

The entries for the Ppat_rep data type shall be stored such that the colour index at the (i,j)-th entry of a DX×DY array of colour indices is given by:

```
colr_ind(i,j) = colr_rect.colr_array[i + DX*j];
```

where colr_rect is of type Ppat_rep and

```
colr_rect.dims.size_x = DX;
colr_rect.dims.size_y = DY;
                    i = 0,...,DX-1;
                    j = 0,...,DY-1;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4 Tables

4.1 Abbreviation policy for construction of identifiers

The following policy is used to construct the identifiers for data types, structure fields, functions, and macros:

a) All identifiers in this part of ISO/IEC 9593 are abbreviated using the same abbreviations for every component and using underscores to denote blanks.

b) The plural of an expression is constructed by adding an "s" after its abbreviation; so, for example, "vector" is abbreviated to "vec" and "vectors" is abbreviated to "vecs"; if an expression is mapped to NULL then its plural is also abbreviated to NULL.

c) Digits are preceeded by underscores, except when the 3D version of a function or type is denoted, then no underscore is used.

d) The following are exceptions to the preceeding rules:

   1) The phrases "polyline", "polymarker", "fill area", "fill area set", and "annotation text" are not abbreviated when they are used in the primitive functions POLYLINE(3), POLYMARKER(3), FILL AREA(3), FILL AREA SET(3), and ANNOTATION TEXT RELATIVE(3).

   2) The phrase "generalized drawing primitives" is abbreviated "gdp".

   3) The phrase "generalized structure elements" is abbreviated "gse".

   4) The phrase "length" is not deleted when forming the macro identifiers for error numbers.

e) Construction of identifiers:

   1) function names:
      "p" (lower case) followed by abbreviated function name in lower case;

   2) data types:
      "P" (upper case) followed by abbreviated data type name in lower case;

   3) fields of data types:
      The name of a field comes from the description of the abstract parameter in PHIGS;

   4) Function macros:
      "Pfn_" followed by abbreviation of function names in lower case;

   5) Error macros:
      "PE_" followed by some abbreviated expression in upper case;

   6) Fields of enumeration types:
      "P" (upper case) followed by a prefix followed by an abbreviation of the field names; this prefix is constant for each enumeration field; all the fields are in upper case.

4.2 Table of abbreviations

Only the words which are abbreviated are listed in table 1. This part of ISO/IEC 9593 abbreviates the words which are in data type names, function names, data type fields names, formal parameter names, and macro names in order to form identifier names. The word "NULL" denotes those words which are deleted completely when forming identifier names.

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
alignment	align
ancestors	ances
and	NULL
annotation	anno

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
annotation text	anno
application	appl
archive	ar
aspect source flag	asf
at	NULL
attributes	attrs
availability	avail
available	avail
between	NULL
category	cat
character	char
cie l*u*v*	cieluv
classification	class
clipping	clip
colour	colr
composition	compose
concatenate	concat
conditional	cond
conflict	conf
conflicting	conf
connection	conn
control	ctrl
current	cur
default	def
deferral	defer
delete	del
dependency	dep
dependent	dep
descendants	descs
device	NULL
devices	NULL
device coordinate	dc
direction	dir
display	disp
dynamic	dyn
dynamics	dyns
element	elem
elements	elems
error	err
evaluate	eval
exclusion	excl
execute	exec
expansion	expan
facilities	facs
factor	NULL
from	NULL
generalized drawing primitive	gdp
generalized drawing primitives	gdp
generalized structure element	gse
generalized structure elements	gse
handling	hand

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
height	ht
highlighting	highl
horizontal	hor
hue lightness saturation	hls
hue saturation value	hsv
identifier	id
identifiers	ids
in	NULL
inclusion	incl
incremental	incr
independent	indep
index	ind
indicator	ind
indices	inds
individual	indiv
initialize	init
input	in
inquire	inq
integer	int
interior	int
invisibility	invis
length	NULL
lengths	NULL
locator	loc
logging	log
logical	NULL
luminance	lum
mapping	map
maximum	max
memory	mem
metafile	NULL
minimum	min
modelling	model
modification	mod
network	net
networks	nets
nominal	nom
normal	norm
number	num
of	NULL
operating	op
orientation	ori
output	out
overflow	overf
parallelogram	paral
pattern	pat
pending	pend
perspective	perspect
pointer	ptr
polyline	line
polymarker	marker

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Table 1 - Abbreviations ordered alphabetically

Word or Phrase	Abbreviation
precision	prec
predefined	pred
primary	prim
priorities	pris
priority	pri
projection	proj
prompt	pr
queue	NULL
record	NULL
rectangle	rect
red green blue	rgb
reference	ref
references	refs
regeneration	regen
relative	rel
representation	rep
request	req
resolution	res
retrieve	ret
saturation	satur
scale factor	NULL
set of	NULL
simultaneous	simult
spacing	space
spatial	spa
state	st
structure	struct
structures	structs
supported	NULL
surface	surf
system	sys
to	NULL
transform	tran
transformation	tran
update	upd
used	NULL
valuator	val
value	NULL
vector	vec
vectors	vecs
vertical	vert
viewport	vp
volume	vol
which	NULL
window	win
workstation	ws
workstations	wss

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

4.3 Function names

4.3.1 List ordered alphabetically by bound name

Note – The PHIGS function INQUIRE HLHSR FACILITIES has been split into two functions, INQUIRE HLHSR MODE FACILITIES and INQUIRE HLHSR IDENTIFIER FACILITES.

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
padd_names_set	ADD NAMES TO SET
panno_text_rel	ANNOTATION TEXT RELATIVE
panno_text_rel3	ANNOTATION TEXT RELATIVE 3
pappl_data	APPLICATION DATA
par_all_structs	ARCHIVE ALL STRUCTURES
par_structs	ARCHIVE STRUCTURES
par_struct_nets	ARCHIVE STRUCTURE NETWORKS
pawait_event	AWAIT EVENT
pbuild_tran_matrix	BUILD TRANSFORMATION MATRIX
pbuild_tran_matrix3	BUILD TRANSFORMATION MATRIX 3
pcell_array	CELL ARRAY
pcell_array3	CELL ARRAY 3
pchange_struct_id	CHANGE STRUCTURE IDENTIFIER
pchange_struct_id_refs	CHANGE STRUCTURE IDENTIFIER AND REFERENCES
pchange_struct_refs	CHANGE STRUCTURE REFERENCES
pclose_ar_file	CLOSE ARCHIVE FILE
pclose_phigs	CLOSE PHIGS
pclose_struct	CLOSE STRUCTURE
pclose_ws	CLOSE WORKSTATION
pcompose_matrix	COMPOSE MATRIX
pcompose_matrix3	COMPOSE MATRIX 3
pcompose_tran_matrix	COMPOSE TRANSFORMATION MATRIX
pcompose_tran_matrix3	COMPOSE TRANSFORMATION MATRIX 3
pcopy_all_elems_struct	COPY ALL ELEMENTS FROM STRUCTURE
pcreate_store	CREATE STORE
pdel_all_structs	DELETE ALL STRUCTURES
pdel_all_structs_ar	DELETE ALL STRUCTURES FROM ARCHIVE
pdel_elem	DELETE ELEMENT
pdel_elems_labels	DELETE ELEMENTS BETWEEN LABELS
pdel_elem_range	DELETE ELEMENT RANGE
pdel_store	DELETE STORE
pdel_struct	DELETE STRUCTURE
pdel_structs_ar	DELETE STRUCTURES FROM ARCHIVE
pdel_struct_net	DELETE STRUCTURE NETWORK
pdel_struct_nets_ar	DELETE STRUCTURE NETWORKS FROM ARCHIVE
pelem_search	ELEMENT SEARCH
pemergency_close_phigs	EMERGENCY CLOSE PHIGS
pempty_struct	EMPTY STRUCTURE
perr_hand	ERROR HANDLING
perr_log	ERROR LOGGING
pescape	ESCAPE
peval_view_map_matrix	EVALUATE VIEW MAPPING MATRIX
peval_view_map_matrix3	EVALUATE VIEW MAPPING MATRIX 3
peval_view_ori_matrix	EVALUATE VIEW ORIENTATION MATRIX

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
peval_view_ori_matrix3	EVALUATE VIEW ORIENTATION MATRIX 3
pexec_struct	EXECUTE STRUCTURE
pfill_area	FILL AREA
pfill_area3	FILL AREA 3
pfill_area_set	FILL AREA SET
pfill_area_set3	FILL AREA SET 3
pflush_events	FLUSH DEVICE EVENTS
pgdp	GENERALIZED DRAWING PRIMITIVE
pgdp3	GENERALIZED DRAWING PRIMITIVE 3
pget_choice	GET CHOICE
pget_item_type	GET ITEM TYPE FROM METAFILE
pget_loc	GET LOCATOR
pget_loc3	GET LOCATOR 3
pget_pick	GET PICK
pget_string	GET STRING
pget_stroke	GET STROKE
pget_stroke3	GET STROKE 3
pget_val	GET VALUATOR
pgse	GENERALIZED STRUCTURE ELEMENT
pincr_spa_search	INCREMENTAL SPATIAL SEARCH
pincr_spa_search3	INCREMENTAL SPATIAL SEARCH 3
pinit_choice	INITIALIZE CHOICE
pinit_choice3	INITIALIZE CHOICE 3
pinit_loc	INITIALIZE LOCATOR
pinit_loc3	INITIALIZE LOCATOR 3
pinit_pick	INITIALIZE PICK
pinit_pick3	INITIALIZE PICK 3
pinit_string	INITIALIZE STRING
pinit_string3	INITIALIZE STRING 3
pinit_stroke	INITIALIZE STROKE
pinit_stroke3	INITIALIZE STROKE 3
pinit_val	INITIALIZE VALUATOR
pinit_val3	INITIALIZE VALUATOR 3
pinq_all_conf_structs	INQUIRE ALL CONFLICTING STRUCTURES
pinq_anno_facs	INQUIRE ANNOTATION FACILITIES
pinq_ar_files	INQUIRE ARCHIVE FILES
pinq_ar_st	INQUIRE ARCHIVE STATE VALUE
pinq_choice_st	INQUIRE CHOICE DEVICE STATE
pinq_choice_st3	INQUIRE CHOICE DEVICE STATE 3
pinq_colr_facs	INQUIRE COLOUR FACILITIES
pinq_colr_model	INQUIRE COLOUR MODEL
pinq_colr_model_facs	INQUIRE COLOUR MODEL FACILITIES
pinq_colr_rep	INQUIRE COLOUR REPRESENTATION
pinq_conf_res	INQUIRE CONFLICT RESOLUTION
pinq_conf_structs_net	INQUIRE CONFLICTING STRUCTURES IN NETWORK
pinq_cur_elem_content	INQUIRE CURRENT ELEMENT CONTENT
pinq_cur_elem_type_size	INQUIRE CURRENT ELEMENT TYPE AND SIZE
pinq_def_choice_data	INQUIRE DEFAULT CHOICE DEVICE DATA
pinq_def_choice_data3	INQUIRE DEFAULT CHOICE DEVICE DATA 3
pinq_def_disp_upd_st	INQUIRE DEFAULT DISPLAY UPDATE STATE
pinq_def_loc_data	INQUIRE DEFAULT LOCATOR DEVICE DATA

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
pinq_def_loc_data3	INQUIRE DEFAULT LOCATOR DEVICE DATA 3
pinq_def_pick_data	INQUIRE DEFAULT PICK DEVICE DATA
pinq_def_pick_data3	INQUIRE DEFAULT PICK DEVICE DATA 3
pinq_def_string_data	INQUIRE DEFAULT STRING DEVICE DATA
pinq_def_string_data3	INQUIRE DEFAULT STRING DEVICE DATA 3
pinq_def_stroke_data	INQUIRE DEFAULT STROKE DEVICE DATA
pinq_def_stroke_data3	INQUIRE DEFAULT STROKE DEVICE DATA 3
pinq_def_val_data	INQUIRE DEFAULT VALUATOR DEVICE DATA
pinq_def_val_data3	INQUIRE DEFAULT VALUATOR DEVICE DATA 3
pinq_disp_space_size	INQUIRE DISPLAY SPACE SIZE
pinq_disp_space_size3	INQUIRE DISPLAY SPACE SIZE 3
pinq_disp_upd_st	INQUIRE DISPLAY UPDATE STATE
pinq_dyns_structs	INQUIRE DYNAMICS OF STRUCTURES
pinq_dyns_ws_attrs	INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES
pinq_edge_facs	INQUIRE EDGE FACILITIES
pinq_edge_rep	INQUIRE EDGE REPRESENTATION
pinq_edit_mode	INQUIRE EDIT MODE
pinq_elem_content	INQUIRE ELEMENT CONTENT
pinq_elem_ptr	INQUIRE ELEMENT POINTER
pinq_elem_type_size	INQUIRE ELEMENT TYPE AND SIZE
pinq_err_hand_mode	INQUIRE ERROR HANDLING MODE
pinq_gdp	INQUIRE GENERALIZED DRAWING PRIMITIVE
pinq_gdp3	INQUIRE GENERALIZED DRAWING PRIMITIVE 3
pinq_gse_facs	INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES
pinq_highl_filter	INQUIRE HIGHLIGHTING FILTER
pinq_hlhsr_id_facs	INQUIRE HLHSR IDENTIFIER FACILITIES
pinq_hlhsr_mode	INQUIRE HLHSR MODE
pinq_hlhsr_mode_facs	INQUIRE HLHSR MODE FACILITIES
pinq_int_facs	INQUIRE INTERIOR FACILITIES
pinq_int_rep	INQUIRE INTERIOR REPRESENTATION
pinq_invis_filter	INQUIRE INVISIBILITY FILTER
pinq_in_overf	INQUIRE INPUT QUEUE OVERFLOW
pinq_line_facs	INQUIRE POLYLINE FACILITIES
pinq_line_rep	INQUIRE POLYLINE REPRESENTATION
pinq_list_avail_gdp	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES
pinq_list_avail_gdp3	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3
pinq_list_avail_gse	INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS
pinq_list_avail_ws_types	INQUIRE LIST OF AVAILABLE WORKSTATION TYPES
pinq_list_colr_inds	INQUIRE LIST OF COLOUR INDICES
pinq_list_edge_inds	INQUIRE LIST OF EDGE INDICES
pinq_list_int_inds	INQUIRE LIST OF INTERIOR INDICES
pinq_list_line_inds	INQUIRE LIST OF POLYLINE INDICES
pinq_list_marker_inds	INQUIRE LIST OF POLYMARKER INDICES
pinq_list_pat_inds	INQUIRE LIST OF PATTERN INDICES
pinq_list_text_inds	INQUIRE LIST OF TEXT INDICES
pinq_list_view_inds	INQUIRE LIST OF VIEW INDICES
pinq_loc_st	INQUIRE LOCATOR DEVICE STATE
pinq_loc_st3	INQUIRE LOCATOR DEVICE STATE 3
pinq_marker_facs	INQUIRE POLYMARKER FACILITIES
pinq_marker_rep	INQUIRE POLYMARKER REPRESENTATION
pinq_model_clip_facs	INQUIRE MODELLING CLIPPING FACILITIES

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
pinq_more_simult_events	INQUIRE MORE SIMULTANEOUS EVENTS
pinq_num_avail_in	INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES
pinq_num_disp_pris	INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED
pinq_open_struct	INQUIRE OPEN STRUCTURE
pinq_open_wss	INQUIRE SET OF OPEN WORKSTATIONS
pinq_paths_ances	INQUIRE PATHS TO ANCESTORS
pinq_paths_descs	INQUIRE PATHS TO DESCENDANTS
pinq_pat_facs	INQUIRE PATTERN FACILITIES
pinq_pat_rep	INQUIRE PATTERN REPRESENTATION
pinq_phigs_facs	INQUIRE PHIGS FACILITIES
pinq_pick_st	INQUIRE PICK DEVICE STATE
pinq_pick_st3	INQUIRE PICK DEVICE STATE 3
pinq_posted_structs	INQUIRE POSTED STRUCTURES
pinq_pred_colr_rep	INQUIRE PREDEFINED COLOUR REPRESENTATION
pinq_pred_edge_rep	INQUIRE PREDEFINED EDGE REPRESENTATION
pinq_pred_int_rep	INQUIRE PREDEFINED INTERIOR REPRESENTATION
pinq_pred_line_rep	INQUIRE PREDEFINED POLYLINE REPRESENTATION
pinq_pred_marker_rep	INQUIRE PREDEFINED POLYMARKER REPRESENTATION
pinq_pred_pat_rep	INQUIRE PREDEFINED PATTERN REPRESENTATION
pinq_pred_text_rep	INQUIRE PREDEFINED TEXT REPRESENTATION
pinq_pred_view_rep	INQUIRE PREDEFINED VIEW REPRESENTATION
pinq_string_st	INQUIRE STRING DEVICE STATE
pinq_string_st3	INQUIRE STRING DEVICE STATE 3
pinq_stroke_st	INQUIRE STROKE DEVICE STATE
pinq_stroke_st3	INQUIRE STROKE DEVICE STATE 3
pinq_struct_ids	INQUIRE STRUCTURE IDENTIFIERS
pinq_struct_st	INQUIRE STRUCTURE STATE VALUE
pinq_struct_status	INQUIRE STRUCTURE STATUS
pinq_sys_st	INQUIRE SYSTEM STATE VALUE
pinq_text_extent	INQUIRE TEXT EXTENT
pinq_text_facs	INQUIRE TEXT FACILITIES
pinq_text_rep	INQUIRE TEXT REPRESENTATION
pinq_val_st	INQUIRE VALUATOR DEVICE STATE
pinq_val_st3	INQUIRE VALUATOR DEVICE STATE 3
pinq_view_facs	INQUIRE VIEW FACILITIES
pinq_view_rep	INQUIRE VIEW REPRESENTATION
pinq_wss_posted	INQUIRE SET OF WORKSTATIONS TO WHICH POSTED
pinq_ws_cat	INQUIRE WORKSTATION CATEGORY
pinq_ws_class	INQUIRE WORKSTATION CLASSIFICATION
pinq_ws_conn_type	INQUIRE WORKSTATION CONNECTION AND TYPE
pinq_ws_st	INQUIRE WORKSTATION STATE VALUE
pinq_ws_st_table	INQUIRE WORKSTATION STATE TABLE LENGTHS
pinq_ws_tran	INQUIRE WORKSTATION TRANSFORMATION
pinq_ws_tran3	INQUIRE WORKSTATION TRANSFORMATION 3
pinterpret_item	INTERPRET ITEM
plabel	LABEL
pmessage	MESSAGE
poffset_elem_ptr	OFFSET ELEMENT POINTER
popen_ar_file	OPEN ARCHIVE FILE
popen_phigs	OPEN PHIGS
popen_struct	OPEN STRUCTURE

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
popen_ws	OPEN WORKSTATION
ppolyline	POLYLINE
ppolyline3	POLYLINE 3
ppolymarker	POLYMARKER
ppolymarker3	POLYMARKER 3
ppost_struct	POST STRUCTURE
pread_item	READ ITEM FROM METAFILE
predraw_all_structs	REDRAW ALL STRUCTURES
premove_names_set	REMOVE NAMES FROM SET
preq_choice	REQUEST CHOICE
preq_loc	REQUEST LOCATOR
preq_loc3	REQUEST LOCATOR 3
preq_pick	REQUEST PICK
preq_string	REQUEST STRING
preq_stroke	REQUEST STROKE
preq_stroke3	REQUEST STROKE 3
preq_val	REQUEST VALUATOR
prestore_model_clip_vol	RESTORE MODELLING CLIPPING VOLUME
pret_all_structs	RETRIEVE ALL STRUCTURES
pret_paths_ances	RETRIEVE PATHS TO ANCESTORS
pret_paths_descs	RETRIEVE PATHS TO DESCENDANTS
pret_structs	RETRIEVE STRUCTURES
pret_struct_ids	RETRIEVE STRUCTURE IDENTIFIERS
pret_struct_nets	RETRIEVE STRUCTURE NETWORKS
protate	ROTATE
protate_x	ROTATE X
protate_y	ROTATE Y
protate_z	ROTATE Z
psample_choice	SAMPLE CHOICE
psample_loc	SAMPLE LOCATOR
psample_loc3	SAMPLE LOCATOR 3
psample_pick	SAMPLE PICK
psample_string	SAMPLE STRING
psample_stroke	SAMPLE STROKE
psample_stroke3	SAMPLE STROKE 3
psample_val	SAMPLE VALUATOR
pscale	SCALE
pscale3	SCALE 3
pset_anno_align	SET ANNOTATION TEXT ALIGNMENT
pset_anno_char_ht	SET ANNOTATION TEXT CHARACTER HEIGHT
pset_anno_char_up_vec	SET ANNOTATION TEXT CHARACTER UP VECTOR
pset_anno_path	SET ANNOTATION TEXT PATH
pset_anno_style	SET ANNOTATION STYLE
pset_char_expan	SET CHARACTER EXPANSION FACTOR
pset_char_ht	SET CHARACTER HEIGHT
pset_char_space	SET CHARACTER SPACING
pset_char_up_vec	SET CHARACTER UP VECTOR
pset_choice_mode	SET CHOICE MODE
pset_colr_model	SET COLOUR MODEL
pset_colr_rep	SET COLOUR REPRESENTATION
pset_conf_res	SET CONFLICT RESOLUTION

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
pset_disp_upd_st	SET DISPLAY UPDATE STATE
pset_edgetype	SET EDGETYPE
pset_edgewidth	SET EDGEWIDTH SCALE FACTOR
pset_edge_colr_ind	SET EDGE COLOUR INDEX
pset_edge_flag	SET EDGE FLAG
pset_edge_ind	SET EDGE INDEX
pset_edge_rep	SET EDGE REPRESENTATION
pset_edit_mode	SET EDIT MODE
pset_elem_ptr	SET ELEMENT POINTER
pset_elem_ptr_label	SET ELEMENT POINTER AT LABEL
pset_err_hand	SET ERROR HANDLING
pset_err_hand_mode	SET ERROR HANDLING MODE
pset_global_tran	SET GLOBAL TRANSFORMATION
pset_global_tran3	SET GLOBAL TRANSFORMATION 3
pset_highl_filter	SET HIGHLIGHTING FILTER
pset_hlhsr_id	SET HLHSR IDENTIFIER
pset_hlhsr_mode	SET HLHSR MODE
pset_indiv_asf	SET INDIVIDUAL ASF
pset_int_colr_ind	SET INTERIOR COLOUR INDEX
pset_int_ind	SET INTERIOR INDEX
pset_int_rep	SET INTERIOR REPRESENTATION
pset_int_style	SET INTERIOR STYLE
pset_int_style_ind	SET INTERIOR STYLE INDEX
pset_invis_filter	SET INVISIBILITY FILTER
pset_linetype	SET LINETYPE
pset_linewidth	SET LINEWIDTH SCALE FACTOR
pset_line_colr_ind	SET POLYLINE COLOUR INDEX
pset_line_ind	SET POLYLINE INDEX
pset_line_rep	SET POLYLINE REPRESENTATION
pset_local_tran	SET LOCAL TRANSFORMATION
pset_local_tran3	SET LOCAL TRANSFORMATION 3
pset_loc_mode	SET LOCATOR MODE
pset_marker_colr_ind	SET POLYMARKER COLOUR INDEX
pset_marker_ind	SET POLYMARKER INDEX
pset_marker_rep	SET POLYMARKER REPRESENTATION
pset_marker_size	SET MARKER SIZE SCALE FACTOR
pset_marker_type	SET MARKER TYPE
pset_model_clip_ind	SET MODELLING CLIPPING INDICATOR
pset_model_clip_vol	SET MODELLING CLIPPING VOLUME
pset_model_clip_vol3	SET MODELLING CLIPPING VOLUME 3
pset_pat_ref_point	SET PATTERN REFERENCE POINT
pset_pat_ref_point_vecs	SET PATTERN REFERENCE POINT AND VECTORS
pset_pat_rep	SET PATTERN REPRESENTATION
pset_pat_size	SET PATTERN SIZE
pset_pick_filter	SET PICK FILTER
pset_pick_id	SET PICK IDENTIFIER
pset_pick_mode	SET PICK MODE
pset_string_mode	SET STRING MODE
pset_stroke_mode	SET STROKE MODE
pset_text_align	SET TEXT ALIGNMENT
pset_text_colr_ind	SET TEXT COLOUR INDEX

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 2 - Function names ordered by bound name

C Name	PHIGS Name
pset_text_font	SET TEXT FONT
pset_text_ind	SET TEXT INDEX
pset_text_path	SET TEXT PATH
pset_text_prec	SET TEXT PRECISION
pset_text_rep	SET TEXT REPRESENTATION
pset_val_mode	SET VALUATOR MODE
pset_view_ind	SET VIEW INDEX
pset_view_rep	SET VIEW REPRESENTATION
pset_view_rep3	SET VIEW REPRESENTATION 3
pset_view_tran_in_pri	SET VIEW TRANSFORMATION INPUT PRIORITY
pset_ws_vp	SET WORKSTATION VIEWPORT
pset_ws_vp3	SET WORKSTATION VIEWPORT 3
pset_ws_win	SET WORKSTATION WINDOW
pset_ws_win3	SET WORKSTATION WINDOW 3
ptext	TEXT
ptext3	TEXT 3
ptranslate	TRANSLATE
ptranslate3	TRANSLATE 3
ptran_point	TRANSFORM POINT
ptran_point3	TRANSFORM POINT 3
punpost_all_structs	UNPOST ALL STRUCTURES
punpost_struct	UNPOST STRUCTURE
pupd_ws	UPDATE WORKSTATION
pwrite_item	WRITE ITEM TO METAFILE

4.3.2 List ordered alphabetically by PHIGS function name

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
ADD NAMES TO SET	padd_names_set
ANNOTATION TEXT RELATIVE	panno_text_rel
ANNOTATION TEXT RELATIVE 3	panno_text_rel3
APPLICATION DATA	pappl_data
ARCHIVE ALL STRUCTURES	par_all_structs
ARCHIVE STRUCTURE NETWORKS	par_struct_nets
ARCHIVE STRUCTURES	par_structs
AWAIT EVENT	pawait_event
BUILD TRANSFORMATION MATRIX	pbuild_tran_matrix
BUILD TRANSFORMATION MATRIX 3	pbuild_tran_matrix3
CELL ARRAY	pcell_array
CELL ARRAY 3	pcell_array3
CHANGE STRUCTURE IDENTIFIER	pchange_struct_id
CHANGE STRUCTURE IDENTIFIER AND REFERENCES	pchange_struct_id_refs
CHANGE STRUCTURE REFERENCES	pchange_struct_refs
CLOSE ARCHIVE FILE	pclose_ar_file
CLOSE PHIGS	pclose_phigs
CLOSE STRUCTURE	pclose_struct
CLOSE WORKSTATION	pclose_ws

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
COMPOSE MATRIX	pcompose_matrix
COMPOSE MATRIX 3	pcompose_matrix3
COMPOSE TRANSFORMATION MATRIX	pcompose_tran_matrix
COMPOSE TRANSFORMATION MATRIX 3	pcompose_tran_matrix3
COPY ALL ELEMENTS FROM STRUCTURE	pcopy_all_elems_struct
CREATE STORE	pcreate_store
DELETE ALL STRUCTURES	pdel_all_structs
DELETE ALL STRUCTURES FROM ARCHIVE	pdel_all_structs_ar
DELETE ELEMENT	pdel_elem
DELETE ELEMENT RANGE	pdel_elem_range
DELETE ELEMENTS BETWEEN LABELS	pdel_elems_labels
DELETE STORE	pdel_store
DELETE STRUCTURE	pdel_struct
DELETE STRUCTURE NETWORK	pdel_struct_net
DELETE STRUCTURE NETWORKS FROM ARCHIVE	pdel_struct_nets_ar
DELETE STRUCTURES FROM ARCHIVE	pdel_structs_ar
ELEMENT SEARCH	pelem_search
EMERGENCY CLOSE PHIGS	pemergency_close_phigs
EMPTY STRUCTURE	pempty_struct
ERROR HANDLING	perr_hand
ERROR LOGGING	perr_log
ESCAPE	pescape
EVALUATE VIEW MAPPING MATRIX	peval_view_map_matrix
EVALUATE VIEW MAPPING MATRIX 3	peval_view_map_matrix3
EVALUATE VIEW ORIENTATION MATRIX	peval_view_ori_matrix
EVALUATE VIEW ORIENTATION MATRIX 3	peval_view_ori_matrix3
EXECUTE STRUCTURE	pexec_struct
FILL AREA	pfill_area
FILL AREA 3	pfill_area3
FILL AREA SET	pfill_area_set
FILL AREA SET 3	pfill_area_set3
FLUSH DEVICE EVENTS	pflush_events
GENERALIZED DRAWING PRIMITIVE	pgdp
GENERALIZED DRAWING PRIMITIVE 3	pgdp3
GENERALIZED STRUCTURE ELEMENT	pgse
GET CHOICE	pget_choice
GET ITEM TYPE FROM METAFILE	pget_item_type
GET LOCATOR	pget_loc
GET LOCATOR 3	pget_loc3
GET PICK	pget_pick
GET STRING	pget_string
GET STROKE	pget_stroke
GET STROKE 3	pget_stroke3
GET VALUATOR	pget_val
INCREMENTAL SPATIAL SEARCH	pincr_spa_search
INCREMENTAL SPATIAL SEARCH 3	pincr_spa_search3
INITIALIZE CHOICE	pinit_choice
INITIALIZE CHOICE 3	pinit_choice3
INITIALIZE LOCATOR	pinit_loc
INITIALIZE LOCATOR 3	pinit_loc3
INITIALIZE PICK	pinit_pick

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
INITIALIZE PICK 3	pinit_pick3
INITIALIZE STRING	pinit_string
INITIALIZE STRING 3	pinit_string3
INITIALIZE STROKE	pinit_stroke
INITIALIZE STROKE 3	pinit_stroke3
INITIALIZE VALUATOR	pinit_val
INITIALIZE VALUATOR 3	pinit_val3
INQUIRE ALL CONFLICTING STRUCTURES	pinq_all_conf_structs
INQUIRE ANNOTATION FACILITIES	pinq_anno_facs
INQUIRE ARCHIVE FILES	pinq_ar_files
INQUIRE ARCHIVE STATE VALUE	pinq_ar_st
INQUIRE CHOICE DEVICE STATE	pinq_choice_st
INQUIRE CHOICE DEVICE STATE 3	pinq_choice_st3
INQUIRE COLOUR FACILITIES	pinq_colr_facs
INQUIRE COLOUR MODEL	pinq_colr_model
INQUIRE COLOUR MODEL FACILITIES	pinq_colr_model_facs
INQUIRE COLOUR REPRESENTATION	pinq_colr_rep
INQUIRE CONFLICT RESOLUTION	pinq_conf_res
INQUIRE CONFLICTING STRUCTURES IN NETWORK	pinq_conf_structs_net
INQUIRE CURRENT ELEMENT CONTENT	pinq_cur_elem_content
INQUIRE CURRENT ELEMENT TYPE AND SIZE	pinq_cur_elem_type_size
INQUIRE DEFAULT CHOICE DEVICE DATA	pinq_def_choice_data
INQUIRE DEFAULT CHOICE DEVICE DATA 3	pinq_def_choice_data3
INQUIRE DEFAULT DISPLAY UPDATE STATE	pinq_def_disp_upd_st
INQUIRE DEFAULT LOCATOR DEVICE DATA	pinq_def_loc_data
INQUIRE DEFAULT LOCATOR DEVICE DATA 3	pinq_def_loc_data3
INQUIRE DEFAULT PICK DEVICE DATA	pinq_def_pick_data
INQUIRE DEFAULT PICK DEVICE DATA 3	pinq_def_pick_data3
INQUIRE DEFAULT STRING DEVICE DATA	pinq_def_string_data
INQUIRE DEFAULT STRING DEVICE DATA 3	pinq_def_string_data3
INQUIRE DEFAULT STROKE DEVICE DATA	pinq_def_stroke_data
INQUIRE DEFAULT STROKE DEVICE DATA 3	pinq_def_stroke_data3
INQUIRE DEFAULT VALUATOR DEVICE DATA	pinq_def_val_data
INQUIRE DEFAULT VALUATOR DEVICE DATA 3	pinq_def_val_data3
INQUIRE DISPLAY SPACE SIZE	pinq_disp_space_size
INQUIRE DISPLAY SPACE SIZE 3	pinq_disp_space_size3
INQUIRE DISPLAY UPDATE STATE	pinq_disp_upd_st
INQUIRE DYNAMICS OF STRUCTURES	pinq_dyns_structs
INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES	pinq_dyns_ws_attrs
INQUIRE EDGE FACILITIES	pinq_edge_facs
INQUIRE EDGE REPRESENTATION	pinq_edge_rep
INQUIRE EDIT MODE	pinq_edit_mode
INQUIRE ELEMENT CONTENT	pinq_elem_content
INQUIRE ELEMENT POINTER	pinq_elem_ptr
INQUIRE ELEMENT TYPE AND SIZE	pinq_elem_type_size
INQUIRE ERROR HANDLING MODE	pinq_err_hand_mode
INQUIRE GENERALIZED DRAWING PRIMITIVE	pinq_gdp
INQUIRE GENERALIZED DRAWING PRIMITIVE 3	pinq_gdp3
INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES	pinq_gse_facs
INQUIRE HIGHLIGHTING FILTER	pinq_highl_filter
INQUIRE HLHSR IDENTIFIER FACILITIES	pinq_hlhsr_id_facs

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
INQUIRE HLHSR MODE	pinq_hlhsr_mode
INQUIRE HLHSR MODE FACILITIES	pinq_hlhsr_mode_facs
INQUIRE INPUT QUEUE OVERFLOW	pinq_in_overf
INQUIRE INTERIOR FACILITIES	pinq_int_facs
INQUIRE INTERIOR REPRESENTATION	pinq_int_rep
INQUIRE INVISIBILITY FILTER	pinq_invis_filter
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES	pinq_list_avail_gdp
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3	pinq_list_avail_gdp3
INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS	pinq_list_avail_gse
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES	pinq_list_avail_ws_types
INQUIRE LIST OF COLOUR INDICES	pinq_list_colr_inds
INQUIRE LIST OF EDGE INDICES	pinq_list_edge_inds
INQUIRE LIST OF INTERIOR INDICES	pinq_list_int_inds
INQUIRE LIST OF PATTERN INDICES	pinq_list_pat_inds
INQUIRE LIST OF POLYLINE INDICES	pinq_list_line_inds
INQUIRE LIST OF POLYMARKER INDICES	pinq_list_marker_inds
INQUIRE LIST OF TEXT INDICES	pinq_list_text_inds
INQUIRE LIST OF VIEW INDICES	pinq_list_view_inds
INQUIRE LOCATOR DEVICE STATE	pinq_loc_st
INQUIRE LOCATOR DEVICE STATE 3	pinq_loc_st3
INQUIRE MODELLING CLIPPING FACILITIES	pinq_model_clip_facs
INQUIRE MORE SIMULTANEOUS EVENTS	pinq_more_simult_events
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES	pinq_num_avail_in
INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED	pinq_num_disp_pris
INQUIRE OPEN STRUCTURE	pinq_open_struct
INQUIRE PATHS TO ANCESTORS	pinq_paths_ances
INQUIRE PATHS TO DESCENDANTS	pinq_paths_descs
INQUIRE PATTERN FACILITIES	pinq_pat_facs
INQUIRE PATTERN REPRESENTATION	pinq_pat_rep
INQUIRE PHIGS FACILITIES	pinq_phigs_facs
INQUIRE PICK DEVICE STATE	pinq_pick_st
INQUIRE PICK DEVICE STATE 3	pinq_pick_st3
INQUIRE POLYLINE FACILITIES	pinq_line_facs
INQUIRE POLYLINE REPRESENTATION	pinq_line_rep
INQUIRE POLYMARKER FACILITIES	pinq_marker_facs
INQUIRE POLYMARKER REPRESENTATION	pinq_marker_rep
INQUIRE POSTED STRUCTURES	pinq_posted_structs
INQUIRE PREDEFINED COLOUR REPRESENTATION	pinq_pred_colr_rep
INQUIRE PREDEFINED EDGE REPRESENTATION	pinq_pred_edge_rep
INQUIRE PREDEFINED INTERIOR REPRESENTATION	pinq_pred_int_rep
INQUIRE PREDEFINED PATTERN REPRESENTATION	pinq_pred_pat_rep
INQUIRE PREDEFINED POLYLINE REPRESENTATION	pinq_pred_line_rep
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	pinq_pred_marker_rep
INQUIRE PREDEFINED TEXT REPRESENTATION	pinq_pred_text_rep
INQUIRE PREDEFINED VIEW REPRESENTATION	pinq_pred_view_rep
INQUIRE SET OF OPEN WORKSTATIONS	pinq_open_wss
INQUIRE SET OF WORKSTATIONS TO WHICH POSTED	pinq_wss_posted
INQUIRE STRING DEVICE STATE	pinq_string_st
INQUIRE STRING DEVICE STATE 3	pinq_string_st3
INQUIRE STROKE DEVICE STATE	pinq_stroke_st
INQUIRE STROKE DEVICE STATE 3	pinq_stroke_st3

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
INQUIRE STRUCTURE IDENTIFIERS	pinq_struct_ids
INQUIRE STRUCTURE STATE VALUE	pinq_struct_st
INQUIRE STRUCTURE STATUS	pinq_struct_status
INQUIRE SYSTEM STATE VALUE	pinq_sys_st
INQUIRE TEXT EXTENT	pinq_text_extent
INQUIRE TEXT FACILITIES	pinq_text_facs
INQUIRE TEXT REPRESENTATION	pinq_text_rep
INQUIRE VALUATOR DEVICE STATE	pinq_val_st
INQUIRE VALUATOR DEVICE STATE 3	pinq_val_st3
INQUIRE VIEW FACILITIES	pinq_view_facs
INQUIRE VIEW REPRESENTATION	pinq_view_rep
INQUIRE WORKSTATION CATEGORY	pinq_ws_cat
INQUIRE WORKSTATION CLASSIFICATION	pinq_ws_class
INQUIRE WORKSTATION CONNECTION AND TYPE	pinq_ws_conn_type
INQUIRE WORKSTATION STATE TABLE LENGTHS	pinq_ws_st_table
INQUIRE WORKSTATION STATE VALUE	pinq_ws_st
INQUIRE WORKSTATION TRANSFORMATION	pinq_ws_tran
INQUIRE WORKSTATION TRANSFORMATION 3	pinq_ws_tran3
INTERPRET ITEM	pinterpret_item
LABEL	plabel
MESSAGE	pmessage
OFFSET ELEMENT POINTER	poffset_elem_ptr
OPEN ARCHIVE FILE	popen_ar_file
OPEN PHIGS	popen_phigs
OPEN STRUCTURE	popen_struct
OPEN WORKSTATION	popen_ws
POLYLINE	ppolyline
POLYLINE 3	ppolyline3
POLYMARKER	ppolymarker
POLYMARKER 3	ppolymarker3
POST STRUCTURE	ppost_struct
READ ITEM FROM METAFILE	pread_item
REDRAW ALL STRUCTURES	predraw_all_structs
REMOVE NAMES FROM SET	premove_names_set
REQUEST CHOICE	preq_choice
REQUEST LOCATOR	preq_loc
REQUEST LOCATOR 3	preq_loc3
REQUEST PICK	preq_pick
REQUEST STRING	preq_string
REQUEST STROKE	preq_stroke
REQUEST STROKE 3	preq_stroke3
REQUEST VALUATOR	preq_val
RESTORE MODELLING CLIPPING VOLUME	prestore_model_clip_vol
RETRIEVE ALL STRUCTURES	pret_all_structs
RETRIEVE PATHS TO ANCESTORS	pret_paths_ances
RETRIEVE PATHS TO DESCENDANTS	pret_paths_descs
RETRIEVE STRUCTURE IDENTIFIERS	pret_struct_ids
RETRIEVE STRUCTURE NETWORKS	pret_struct_nets
RETRIEVE STRUCTURES	pret_structs
ROTATE	protate
ROTATE X	protate_x

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
ROTATE Y	protate_y
ROTATE Z	protate_z
SAMPLE CHOICE	psample_choice
SAMPLE LOCATOR	psample_loc
SAMPLE LOCATOR 3	psample_loc3
SAMPLE PICK	psample_pick
SAMPLE STRING	psample_string
SAMPLE STROKE	psample_stroke
SAMPLE STROKE 3	psample_stroke3
SAMPLE VALUATOR	psample_val
SCALE	pscale
SCALE 3	pscale3
SET ANNOTATION STYLE	pset_anno_style
SET ANNOTATION TEXT ALIGNMENT	pset_anno_align
SET ANNOTATION TEXT CHARACTER HEIGHT	pset_anno_char_ht
SET ANNOTATION TEXT CHARACTER UP VECTOR	pset_anno_char_up_vec
SET ANNOTATION TEXT PATH	pset_anno_path
SET CHARACTER EXPANSION FACTOR	pset_char_expan
SET CHARACTER HEIGHT	pset_char_ht
SET CHARACTER SPACING	pset_char_space
SET CHARACTER UP VECTOR	pset_char_up_vec
SET CHOICE MODE	pset_choice_mode
SET COLOUR MODEL	pset_colr_model
SET COLOUR REPRESENTATION	pset_colr_rep
SET CONFLICT RESOLUTION	pset_conf_res
SET DISPLAY UPDATE STATE	pset_disp_upd_st
SET EDGE COLOUR INDEX	pset_edge_colr_ind
SET EDGE FLAG	pset_edge_flag
SET EDGE INDEX	pset_edge_ind
SET EDGE REPRESENTATION	pset_edge_rep
SET EDGETYPE	pset_edgetype
SET EDGEWIDTH SCALE FACTOR	pset_edgewidth
SET EDIT MODE	pset_edit_mode
SET ELEMENT POINTER	pset_elem_ptr
SET ELEMENT POINTER AT LABEL	pset_elem_ptr_label
SET ERROR HANDLING	pset_err_hand
SET ERROR HANDLING MODE	pset_err_hand_mode
SET GLOBAL TRANSFORMATION	pset_global_tran
SET GLOBAL TRANSFORMATION 3	pset_global_tran3
SET HIGHLIGHTING FILTER	pset_highl_filter
SET HLHSR IDENTIFIER	pset_hlhsr_id
SET HLHSR MODE	pset_hlhsr_mode
SET INDIVIDUAL ASF	pset_indiv_asf
SET INTERIOR COLOUR INDEX	pset_int_colr_ind
SET INTERIOR INDEX	pset_int_ind
SET INTERIOR REPRESENTATION	pset_int_rep
SET INTERIOR STYLE	pset_int_style
SET INTERIOR STYLE INDEX	pset_int_style_ind
SET INVISIBILITY FILTER	pset_invis_filter
SET LINETYPE	pset_linetype
SET LINEWIDTH SCALE FACTOR	pset_linewidth

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table 3 - Function names ordered by PHIGS name

PHIGS Name	C Name
SET LOCAL TRANSFORMATION	pset_local_tran
SET LOCAL TRANSFORMATION 3	pset_local_tran3
SET LOCATOR MODE	pset_loc_mode
SET MARKER SIZE SCALE FACTOR	pset_marker_size
SET MARKER TYPE	pset_marker_type
SET MODELLING CLIPPING INDICATOR	pset_model_clip_ind
SET MODELLING CLIPPING VOLUME	pset_model_clip_vol
SET MODELLING CLIPPING VOLUME 3	pset_model_clip_vol3
SET PATTERN REFERENCE POINT	pset_pat_ref_point
SET PATTERN REFERENCE POINT AND VECTORS	pset_pat_ref_point_vecs
SET PATTERN REPRESENTATION	pset_pat_rep
SET PATTERN SIZE	pset_pat_size
SET PICK FILTER	pset_pick_filter
SET PICK IDENTIFIER	pset_pick_id
SET PICK MODE	pset_pick_mode
SET POLYLINE COLOUR INDEX	pset_line_colr_ind
SET POLYLINE INDEX	pset_line_ind
SET POLYLINE REPRESENTATION	pset_line_rep
SET POLYMARKER COLOUR INDEX	pset_marker_colr_ind
SET POLYMARKER INDEX	pset_marker_ind
SET POLYMARKER REPRESENTATION	pset_marker_rep
SET STRING MODE	pset_string_mode
SET STROKE MODE	pset_stroke_mode
SET TEXT ALIGNMENT	pset_text_align
SET TEXT COLOUR INDEX	pset_text_colr_ind
SET TEXT FONT	pset_text_font
SET TEXT INDEX	pset_text_ind
SET TEXT PATH	pset_text_path
SET TEXT PRECISION	pset_text_prec
SET TEXT REPRESENTATION	pset_text_rep
SET VALUATOR MODE	pset_val_mode
SET VIEW INDEX	pset_view_ind
SET VIEW REPRESENTATION	pset_view_rep
SET VIEW REPRESENTATION 3	pset_view_rep3
SET VIEW TRANSFORMATION INPUT PRIORITY	pset_view_tran_in_pri
SET WORKSTATION VIEWPORT	pset_ws_vp
SET WORKSTATION VIEWPORT 3	pset_ws_vp3
SET WORKSTATION WINDOW	pset_ws_win
SET WORKSTATION WINDOW 3	pset_ws_win3
TEXT	ptext
TEXT 3	ptext3
TRANSFORM POINT	ptran_point
TRANSFORM POINT 3	ptran_point3
TRANSLATE	ptranslate
TRANSLATE 3	ptranslate3
UNPOST ALL STRUCTURES	punpost_all_structs
UNPOST STRUCTURE	punpost_struct
UPDATE WORKSTATION	pupd_ws
WRITE ITEM TO METAFILE	pwrite_item

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577361 780 ■

5 Type definitions

5.1 Mapping of PHIGS data types

ISO/IEC 9592-1 specifies a set of abstract data types. Table 4 gives the mapping from those data types to the data types defined in later this binding.

Table 4 - Mapping of PHIGS data types to C

PHIGS data type		C binding data type
I	integer	`Pint`
R	real	`Pfloat`
S	string	`char *`
P2	2D point	`Ppoint`
P3	3D point	`Ppoint3`
V2	2D vector	`Pvec`
V3	3D vector	`Pvec3`
E	enumeration type	`typedef enum`
NM	classification	`Pint`
FR	filter	`Pfilter`
PP	pick path item	`Ppick_path_elem`
ER	element reference	`Pelem_ref`
HS2	half-space	`Phalf_space`
HS3	half-space	`Phalf_space3`
FP	font/precision pair	not applicable
SE	structure element	not applicable
PS	posted structure	`Pposted_struct`
B	bounding range	2 × B is `Plimit`; 3 × B is `Plimit3`
CLR	colour specification	`Pcolr_rep`
CC	chromaticity coefficient and luminance value	`Pcieluv`
C	connection identifier	`void *`
F	file	`char *`
W	workstation type	`Pint`
MCV	modelling clipping volume	not applicable
G2	2D GDP identifier	`Pint`
G3	3D GDP identifier	`Pint`
GS	GSE identifier	`Pint`
AI	archive file identifier	`Pint`
PI	pick identifier	`Pint`
EI	escape identifier	`Pint`
FN	function name	`Pint`
WI	workstation identifier	`Pint`

5.2 Environmental type definitions

The data types defined in this section allow for the ease of porting PHIGS application programs between different environments. These types are used as the basis for all the other data types.

`Pfloat` — floating point number

This data type must be defined by the implementation as a type suitable for containing the single precision floating point values used by the PHIGS implementation. Suggest:

```
typedef float   Pfloat;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Pint` — integer

This data type must be defined by the implementation as a type suitable for containing the integer values used by the PHIGS implementation. Suggest:

```
typedef int      Pint;
```

5.3 Implementation dependent type definitions

The following type definitions may be modified by the implementation. If the implementation adds new fields to the unions, this convention shall be followed for naming the new field:

"xxx_tn"

where:

xxx is the name of the field. Use "pet" for input device data record prompt echo types, "gdp" for GENERALIZED DRAWING PRIMITIVE data record, "gdp3" for GENERALIZED DRAWING PRIMITIVE 3 data records, "gse" for GENERALIZED STRUCTURE ELEMENT, or "escape_in" for ESCAPE input data records, "escape_out" for ESCAPE output data records.

t is the type of field. Use "r" for fields corresponding registered items and "u" for fields for unregistered items.

n is "n" is the item number; for unregistered items, the absolute value of the item number shall be used.

The contents of the metafile item data record (`Pitem_data`) are not subject to registration and therefore do not follow this naming convention.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pchoice_data CHOICE DATA RECORD

```
typedef struct {
  union Pchoice_pets {
    struct Pchoice_pet_r1 {
          . . .                          /* implementation dependent     */
    } pet_r1;
    struct Pchoice_pet_r2 {
          Pint        num_prompts;       /* number of prompts            */
          Ppr_switch  *prompts;          /* array of prompts             */
          . . .                          /* implementation dependent     */
    } pet_r2;
    struct Pchoice_pet_r3 {
          Pint        num_strings;       /* number of choice strings     */
          char        **strings;         /* array of choice strings      */
          . . .                          /* implementation dependent     */
    } pet_r3;
    struct Pchoice_pet_r4 {
          Pint        num_strings;       /* number of choice strings     */
          char        **strings;         /* array of choice strings      */
          . . .                          /* implementation dependent     */
    } pet_r4;
    struct Pchoice_pet_r5 {
          Pint        struct_id;         /* struct identifier            */
          Pint        num_pick_ids;      /* number of pick identifiers   */
          Pint        *pick_ids;         /* array of pick identifiers    */
          . . .                          /* implementation dependent     */
    } pet_r5;
    . . .                                /* implementation defined PET's */
  } pets;
} Pchoice_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pchoice_data3 CHOICE DATA RECORD 3

```
typedef struct {
  union Pchoice3_pets {
    struct Pchoice3_pet_r1 {
          . . .                       /* implementation dependent     */
    } pet_r1;
    struct Pchoice3_pet_r2 {
          Pint        num_prompts;    /* number of prompts            */
          Ppr_switch  *prompts;       /* array of prompts             */
          . . .                       /* implementation dependent     */
    } pet_r2;
    struct Pchoice3_pet_r3 {
          Pint        num_strings;    /* number of choice strings     */
          char        **strings;      /* array of choice strings      */
          . . .                       /* implementation dependent     */
    } pet_r3;
    struct Pchoice3_pet_r4 {
          Pint        num_strings;    /* number of choice strings     */
          char        **strings;      /* array of choice strings      */
          . . .                       /* implementation dependent     */
    } pet_r4;
    struct Pchoice3_pet_r5 {
          Pint        struct_id;      /* struct identifier            */
          Pint        num_pick_ids;   /* number of pick identifiers   */
          Pint        *pick_ids;      /* array of pick identifiers    */
          . . .                       /* implementation dependent     */
    } pet_r5;
    . . .                             /* implementation defined PET's */
  } pets;
} Pchoice_data3;
```

Pcolr_rep COLOUR REPRESENTATION

```
typedef union {
  Prgb     rgb;        /* Red Green Blue colour specification          */
  Pcieluv  cieluv;     /* CIE L*u*v* colour specification              */
  Phls     hls;        /* Hue Lightness Saturation colour specification */
  Phsv     hsv;        /* Hue Saturation Value colour specification    */
  Pdata    unsupp;     /* colour in an unsupported colour model        */
  . . .                /* implementation defined                       */
} Pcolr_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pescape_in_data ESCAPE INPUT DATA RECORD

```
typedef union {
  struct Pescape_in_r1 {
        . . .       /* registration dependent */
  } escape_in_r1;
  . . .             /* implementation defined */
} Pescape_in_data;
```

Pescape_out_data ESCAPE OUTPUT DATA RECORD

```
typedef union {
  struct Pescape_out_r1 {
        . . .       /* registration dependent */
  } escape_out_r1;
  . . .             /* implementation defined */
} Pescape_out_data;
```

Pgdp_data GDP DATA RECORD

```
typedef union {
  struct Pgdp_r1 {
        . . .         /* registration dependent      */
  } gdp_r1;
  Pdata  unsupp;      /* unsupported GDP data record */
  . . .               /* implementation defined      */
} Pgdp_data;
```

Pgdp_data3 GDP DATA RECORD 3

```
typedef union {
  struct Pgdp3_r1 {
        . . .         /* registration dependent       */
  } gdp3_r1;
  Pdata  unsupp;      /* unsupported GDP3 data record */
  . . .               /* implementation defined       */
} Pgdp_data3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pgse_data GSE DATA RECORD

```
typedef union {
  struct Pgse_rl {
        . . .          /* registration dependent      */
  } gse_rl;
  Pdata  unsupp;       /* unsupported GSE data record */
  . . .                /* implementation defined      */
} Pgse_data;
```

Pitem_data ITEM DATA RECORD

```
typedef union {
  . . .                /* Metafile Records                */
  Pdata  unsupp;       /* unsupported Metafile item data */
} Pitem_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Ploc_data LOCATOR DATA RECORD

```
typedef struct {
  union Ploc_pets {
    struct Ploc_pet_r1 {
          . . .                                          /* impl. dependent       */
    } pet_r1;
    struct Ploc_pet_r2 {
          . . .                                          /* impl. dependent       */
    } pet_r2;
    struct Ploc_pet_r3 {
          . . .                                          /* impl. dependent       */
    } pet_r3;
    struct Ploc_pet_r4 {
          Pline_attrs          line_attrs;               /* polyline attributes   */
          . . .                                          /* impl. dependent       */
    } pet_r4;
    struct Ploc_pet_r5 {
          Pline_fill_ctrl_flag line_fill_ctrl_flag;      /* control flag          */
          union Ploc_attrs {
                               Pline_attrs               line_attrs;
                                                         /* polyline attributes   */
                               Pint_attrs                int_attrs;
                                                         /* interior attributes   */
                               struct Ploc_fill_set {
                                                         Pint_attrs  int_attrs;
                                                         /* interior attributes   */
                                                         Pedge_attrs  edge_attrs;
                                                         /* edge attributes       */
                               } fill_set;
          } attrs;
          . . .                                          /* impl. dependent       */
    } pet_r5;
    struct Ploc_pet_r6 {
          . . .                                          /* impl. dependent       */
    } pet_r6;
    . . .                                                /* impl. defined PET's   */
  } pets;
} Ploc_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Ploc_data3 LOCATOR DATA RECORD 3

```
typedef struct {
  union Ploc3_pets {
    struct Ploc3_pet_r1 {
          . . .                                          /* impl. dependent       */
    } pet_r1;
    struct Ploc3_pet_r2 {
          . . .                                          /* impl. dependent       */
    } pet_r2;
    struct Ploc3_pet_r3 {
          . . .                                          /* impl. dependent       */
    } pet_r3;
    struct Ploc3_pet_r4 {
          Pline_attrs          line_attrs;               /* polyline attributes   */
          . . .                                          /* impl. dependent       */
    } pet_r4;
    struct Ploc3_pet_r5 {
          Pline_fill_ctrl_flag line_fill_ctrl_flag;      /* control flag          */
          union Ploc3_attrs {
                               Pline_attrs               line_attrs;
                                                         /* polyline attributes   */
                               Pint_attrs                int_attrs;
                                                         /* interior attributes   */
                               struct Ploc3_fill_set {
                                                         Pint_attrs  int_attrs;
                                                         /* interior attributes   */
                                                         Pedge_attrs  edge_attrs;
                                                         /* edge attributes       */
                               } fill_set;
          } attrs;
          . . .                                          /* impl. dependent       */
    } pet_r5;
    struct Ploc3_pet_r6 {
          . . .                                          /* impl. dependent       */
    } pet_r6;
    . . .                                                /* impl. defined PET's   */
  } pets;
} Ploc_data3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Ppick_data PICK DATA RECORD

```
typedef struct {
  union Ppick_pets {
    struct Ppick_pet_r1 {
          . . .    /* implementation dependent     */
    } pet_r1;
    . . .          /* implementation defined PET's */
  } pets;
} Ppick_data;
```

Ppick_data3 PICK DATA RECORD 3

```
typedef struct {
  union Ppick3_pets {
    struct Ppick3_pet_r1 {
          . . .    /* implementation dependent     */
    } pet_r1;
    . . .          /* implementation defined PET's */
  } pets;
} Ppick_data3;
```

Pstring_data STRING DATA RECORD

```
typedef struct {
  Pint  in_buf_size;          /* input buffer size            */
  Pint  init_pos;             /* initial editing position     */
  union Pstring_pets {
        struct Pstring_pet_r1 {
                    . . .     /* implementation dependent     */
        } pet_r1;
        . . .                 /* implementation defined PET's */
  } pets;
} Pstring_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pstring_data3 STRING DATA RECORD 3

```
typedef struct {
  Pint  in_buf_size;          /* input buffer size           */
  Pint  init_pos;             /* initial editing position    */
  union Pstring3_pets {
        struct Pstring3_pet_r1 {
                    . . .     /* implementation dependent    */
        } pet_r1;
        . . .                 /* implementation defined PET's */
  } pets;
} Pstring_data3;
```

Pstroke_data STROKE DATA RECORD

```
typedef struct {
  Pint    in_buf_size;                              /* input buffer size           */
  Pint    init_pos;                                 /* initial editing position    */
  Pfloat  x_interval;                               /* x interval                  */
  Pfloat  y_interval;                               /* y interval                  */
  Pfloat  time_interval;                            /* time interval               */
  union Pstroke_pets {
        struct Pstroke_pet_r1 {
                    . . .                           /* implementation dependent    */
        } pet_r1;
        struct Pstroke_pet_r2 {
                    . . .                           /* implementation dependent    */
        } pet_r2;
        struct Pstroke_pet_r3 {
                    Pmarker_attrs  marker_attrs;    /* marker attributes           */
                    . . .                           /* implementation dependent    */
        } pet_r3;
        struct Pstroke_pet_r4 {
                    Pline_attrs    line_attrs;      /* line attributes             */
                    . . .                           /* implementation dependent    */
        } pet_r4;
        . . .                                       /* implementation defined PET's */
  } pets;
} Pstroke_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Pstroke_data3 STROKE DATA RECORD 3

```
typedef struct {
  Pint    in_buf_size;                                  /* input buffer size
  Pint    init_pos;                                     /* initial editing position
  Pfloat  x_interval;                                   /* x interval
  Pfloat  y_interval;                                   /* y interval
  Pfloat  z_interval;                                   /* z interval
  Pfloat  time_interval;                                /* time interval
  union Pstroke3_pets {
        struct Pstroke3_pet_r1 {
                    . . .                               /* implementation dependent
        } pet_r1;
        struct Pstroke3_pet_r2 {
                    . . .                               /* implementation dependent
        } pet_r2;
        struct Pstroke3_pet_r3 {
                    Pmarker_attrs  marker_attrs;        /* marker attributes
                    . . .                               /* implementation dependent
        } pet_r3;
        struct Pstroke3_pet_r4 {
                    Pline_attrs    line_attrs;          /* line attributes
                    . . .                               /* implementation dependent
        } pet_r4;
        . . .                                           /* implementation defined P
  } pets;
} Pstroke_data3;
```

Pval_data VALUATOR DATA RECORD

```
typedef struct {
  Pfloat  low_value;          /* low value of valuator range  */
  Pfloat  high_value;         /* high value of valuator range */
  union Pval_pets {
        struct Pval_pet_r1 {
                  . . .       /* implementation dependent     */
        } pet_r1;
        . . .                 /* implementation defined PET's */
  } pets;
} Pval_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pval_data3 VALUATOR DATA RECORD 3

```
typedef struct {
  Pfloat  low_value;            /* low value of valuator range  */
  Pfloat  high_value;           /* high value of valuator range */
  union Pval3_pets {
        struct Pval3_pet_r1 {
                    . . .       /* implementation dependent     */
        } pet_r1;
        . . .                   /* implementation defined PET's */
  } pets;
} Pval_data3;
```

5.4 Implementation independent type definitions

Par_file ARCHIVE FILE

```
typedef struct {
  Pint   id;          /* archive file identifier */
  char   *name;       /* archive file name       */
} Par_file;
```

Par_file_list ARCHIVE FILE LIST

```
typedef struct {
  Pint      num_ar_files;     /* number of archive files */
  Par_file  *ar_files;        /* list of archive files   */
} Par_file_list;
```

Par_st ARCHIVE STATE

```
typedef enum {
   PST_ARCL,
   PST_AROP
} Par_st;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pasf ASPECT SOURCE FLAG

```
typedef enum {
    PASF_BUNDLED,
    PASF_INDIV
} Pasf;
```

Paspect ASPECT

```
typedef enum {
    PASPECT_LINETYPE,
    PASPECT_LINEWIDTH,
    PASPECT_LINE_COLR_IND,
    PASPECT_MARKER_TYPE,
    PASPECT_MARKER_SIZE,
    PASPECT_MARKER_COLR_IND,
    PASPECT_TEXT_FONT,
    PASPECT_TEXT_PREC,
    PASPECT_CHAR_EXPAN,
    PASPECT_CHAR_SPACE,
    PASPECT_TEXT_COLR_IND,
    PASPECT_INT_STYLE,
    PASPECT_INT_STYLE_IND,
    PASPECT_INT_COLR_IND,
    PASPECT_EDGE_FLAG,
    PASPECT_EDGETYPE,
    PASPECT_EDGEWIDTH,
    PASPECT_EDGE_COLR_IND
} Paspect;
```

Pattrs ATTRIBUTES USED

```
typedef enum {
    PATTR_LINE,
    PATTR_MARKER,
    PATTR_TEXT,
    PATTR_INT,
    PATTR_EDGE
} Pattrs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pcieluv CIE L*U*V*

```
typedef struct {
  Pfloat  cieluv_x;          /* x coefficient */
  Pfloat  cieluv_y;          /* y coefficient */
  Pfloat  cieluv_y_lum;      /* y luminance   */
} Pcieluv;
```

Pclip_ind CLIPPING INDICATOR

```
typedef enum {
    PIND_NO_CLIP,
    PIND_CLIP
} Pclip_ind;
```

Pcolr_avail COLOUR AVAILABILITY

```
typedef enum {
    PAVAIL_MONOCHR,
    PAVAIL_COLR
} Pcolr_avail;
```

Pcolr_facs COLOUR FACILITIES

```
typedef struct {
  Pint         num_colrs;          /* number of colours                    */
  Pcolr_avail  colr_avail;         /* colour availability                  */
  Pint         num_pred_inds;      /* number of predefined colour indices  */
  Pcieluv      prim_colrs[3];      /* primary colours                      */
} Pcolr_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pcompose_type COMPOSITION TYPE

```
typedef enum {
    PTYPE_PRECONCAT,
    PTYPE_POSTCONCAT,
    PTYPE_REPLACE
} Pcompose_type;
```

Pconf_res CONFLICT RESOLUTION

```
typedef enum {
    PRES_MAINTAIN,
    PRES_ABANDON,
    PRES_UPD
} Pconf_res;
```

Pctrl_flag CONTROL FLAG

```
typedef enum {
    PFLAG_COND,
    PFLAG_ALWAYS
} Pctrl_flag;
```

Pdata DATA

```
typedef struct {
  size_t  size;       /* sizeof data     */
  void    *data;      /* pointer to data */
} Pdata;
```

Pdc_units DEVICE COORDINATE UNITS

```
typedef enum {
    PDC_METRES,
    PDC_OTHER
} Pdc_units;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pdefer_mode DEFERRAL MODE

```
typedef enum {
    PDEFER_ASAP,
    PDEFER_BNIG,
    PDEFER_BNIL,
    PDEFER_ASTI,
    PDEFER_WAIT
} Pdefer_mode;
```

Pdisp_space_size DISPLAY SPACE SIZE

```
typedef struct {
  Pdc_units    dc_units;        /* device coordinate units         */
  Pfloat_size  size_dc;         /* device size in coordinate units */
  Pint_size    size_raster;     /* device size in raster units     */
} Pdisp_space_size;
```

Pdisp_space_size3 DISPLAY SPACE SIZE 3

```
typedef struct {
  Pdc_units     dc_units;        /* device coordinate units           */
  Pfloat_size3  size_dc;         /* device volume in coordinate units */
  Pint_size3    size_raster;     /* device volume in raster units     */
} Pdisp_space_size3;
```

Pdisp_surf_empty DISPLAY SURFACE EMPTY

```
typedef enum {
    PSURF_NOT_EMPTY,
    PSURF_EMPTY
} Pdisp_surf_empty;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pdyns_structs DYNAMICS OF STRUCTURES

```
typedef struct {
  Pdyn_mod  content;      /* structure content     */
  Pdyn_mod  post;         /* post structure        */
  Pdyn_mod  unpost;       /* unpost structure      */
  Pdyn_mod  del;          /* delete structure      */
  Pdyn_mod  ref;          /* structure reference   */
} Pdyns_structs;
```

Pdyns_ws_attrs DYNAMICS OF WORKSTATION ATTRIBUTES

```
typedef struct {
  Pdyn_mod  line_bundle;      /* polyline bundle representation   */
  Pdyn_mod  marker_bundle;    /* polymarker bundle representation */
  Pdyn_mod  text_bundle;      /* text bundle representation       */
  Pdyn_mod  int_bundle;       /* interior bundle representation   */
  Pdyn_mod  edge_bundle;      /* edge bundle representation       */
  Pdyn_mod  pat_rep;          /* pattern representation           */
  Pdyn_mod  colr_rep;         /* colour representation            */
  Pdyn_mod  view_rep;         /* view representation              */
  Pdyn_mod  ws_tran;          /* workstation transformation       */
  Pdyn_mod  highl_filter;     /* highlighting filter              */
  Pdyn_mod  invis_filter;     /* invisibility filter              */
  Pdyn_mod  hlhsr_mode;       /* HLHSR mode                       */
} Pdyns_ws_attrs;
```

Pdyn_mod DYNAMIC MODIFICATION

```
typedef enum {
    PDYN_IRG,
    PDYN_IMM,
    PDYN_CBS
} Pdyn_mod;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pecho_switch ECHO SWITCH

```
typedef enum {
    PSWITCH_NO_ECHO,
    PSWITCH_ECHO
} Pecho_switch;
```

Pedge_attrs EDGE ATTRIBUTES

```
typedef struct {
  Pasf          flag_asf;          /* edge flag asf          */
  Pasf          type_asf;          /* edgetype asf           */
  Pasf          width_asf;         /* edgewidth asf          */
  Pasf          colr_ind_asf;      /* edge colour index asf */
  Pint          ind;               /* edge index             */
  Pedge_bundle  bundle;            /* edge bundle            */
} Pedge_attrs;
```

Pedge_bundle EDGE BUNDLE

```
typedef struct {
  Pedge_flag  flag;          /* edge flag              */
  Pint        type;          /* edgetype               */
  Pfloat      width;         /* edgewidth scale factor */
  Pint        colr_ind;      /* edge colour index      */
} Pedge_bundle;
```

Pedge_facs EDGE FACILITIES

```
typedef struct {
  Pint_list  types;             /* list of edgetypes                    */
  Pint       num_widths;        /* number of available edgewidths       */
  Pfloat     nom_width;         /* nominal edgewidth                    */
  Pfloat     min_width;         /* min edgewidth                        */
  Pfloat     max_width;         /* max edgewidth                        */
  Pint       num_pred_inds;     /* number of predefined bundle indices */
} Pedge_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577379 910 ■

Pedge_flag EDGE FLAG

```
typedef enum {
    PEDGE_OFF,
    PEDGE_ON
} Pedge_flag;
```

Pedit_mode EDIT MODE

```
typedef enum {
    PEDIT_INSERT,
    PEDIT_REPLACE
} Pedit_mode;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pelem_data ELEMENT DATA

```
typedef union {
  Pint              int_data;                    /* integer valued data    */
  Pfloat            float_data;                  /* float valued data      */
  Ppoint_list3      point_list3;                 /* list of 3d points      */
  Ppoint_list       point_list;                  /* list of 2d points      */
  Ppoint_list_list3 point_list_list3;            /* list of 3d point lists */
  Ppoint_list_list  point_list_list;             /* list of 2d point lists */
  struct Pelem_text3 {
                    Ppoint3         pos;         /* text position          */
                    Pvec3           dir[2];      /* direction vectors      */
                    char            *char_string; /* character string      */
  } text3;
  struct Pelem_text {
                    Ppoint          pos;         /* text position          */
                    char            *char_string; /* character string      */
  } text;
  struct Pelem_anno_text_rel3 {
                    Ppoint3         ref_point;   /* reference point        */
                    Pvec3           offset;      /* anno. offset           */
                    char            *char_string; /* character string      */
  } anno_text_rel3;
  struct Pelem_anno_text_rel {
                    Ppoint          ref_point;   /* reference point        */
                    Pvec            offset;      /* anno. offset           */
                    char            *char_string; /* character string      */
  } anno_text_rel;
  struct Pelem_cell_array3 {
                    Pparal          paral;       /* parallelogram          */
                    Ppat_rep        colr_array;  /* colour array           */
  } cell_array3;
  struct Pelem_cell_array {
                    Prect           rect;        /* rectangle              */
                    Ppat_rep        colr_array;  /* colour array           */
  } cell_array;
  struct Pelem_gdp3 {
                    Pint            id;          /* GDP3 id                */
                    Ppoint_list3    point_list;  /* point list             */
                    Pgdp_data3      data;        /* data record            */
  } gdp3;
  struct Pelem_gdp {
                    Pint            id;          /* GDP id                 */
                    Ppoint_list     point_list;  /* point list             */
                    Pgdp_data       data;        /* data record            */
  } gdp;
  Ptext_prec        text_prec;                   /* text precision         */
  Pvec              char_up_vec;                 /* char up vector         */
  Ptext_path        text_path;                   /* text path              */
  Ptext_align       text_align;                  /* text alignment         */
  Pint_style        int_style;                   /* interior style         */
  Pedge_flag        edge_flag;                   /* edge flag              */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
    Ppoint              pat_ref_point;                  /* pattern ref. point
    Pfloat_size         pat_size;                       /* pattern size
    struct Pelem_pat_ref_point_vecs {
                        Ppoint3             ref_point;  /* pattern ref. point
                        Pvec3               ref_vec[2]; /* vectors
    } pat_ref_point_vecs;
    Pint_list           names;                          /* name sets
    struct Pelem_asf {
                        Paspect             id;         /* attribute id
                        Pasf                source;     /* attribute source
    } asf;
    struct Pelem_local_tran3 {
                        Pcompose_type       compose_type; /* composition type
                        Pmatrix3            matrix;     /* tran. matrix
    } local_tran3;
    struct Pelem_local_tran {
                        Pcompose_type       compose_type; /* composition type
                        Pmatrix             matrix;     /* tran. matrix
    } local_tran;
    Pmatrix3            global_tran3;                   /* global transform3
    Pmatrix             global_tran;                    /* global transform
    struct Pelem_model_clip3 {
                        Pint                op;         /* operator
                        Phalf_space_list3   half_spaces; /* half space list
    } model_clip3;
    struct Pelem_model_clip {
                        Pint                op;         /* operator
                        Phalf_space_list    half_spaces; /* half space list
    } model_clip;
    Pclip_ind           clip_ind;                       /* clipping indicator
    Pdata               appl_data;                      /* application data
    struct Pelem_gse {
                        Pint                id;         /* GSE id
                        Pgse_data           data;       /* GSE data record
    } gse;
} Pelem_data;
```

Pelem_ref ELEMENT REFERENCE

```
typedef struct {
    Pint  struct_id;     /* structure identifier */
    Pint  elem_pos;      /* element position     */
} Pelem_ref;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

`Pelem_ref_list` ELEMENT REFERENCE LIST

```
typedef struct {
  Pint       num_elem_refs;      /* number of element references */
  Pelem_ref  *elem_refs;         /* list of element references   */
} Pelem_ref_list;
```

`Pelem_ref_list_list` ELEMENT REFERENCE LIST LIST

```
typedef struct {
  Pint            num_elem_ref_lists;     /* number of element references lists */
  Pelem_ref_list  *elem_ref_lists;        /* list of element reference lists    */
} Pelem_ref_list_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pelem_type ELEMENT TYPE

```
typedef enum {
    PELEM_ALL,
    PELEM_NIL,
    PELEM_POLYLINE3,
    PELEM_POLYLINE,
    PELEM_POLYMARKER3,
    PELEM_POLYMARKER,
    PELEM_TEXT3,
    PELEM_TEXT,
    PELEM_ANNO_TEXT_REL3,
    PELEM_ANNO_TEXT_REL,
    PELEM_FILL_AREA3,
    PELEM_FILL_AREA,
    PELEM_FILL_AREA_SET3,
    PELEM_FILL_AREA_SET,
    PELEM_CELL_ARRAY3,
    PELEM_CELL_ARRAY,
    PELEM_GDP3,
    PELEM_GDP,
    PELEM_LINE_IND,
    PELEM_MARKER_IND,
    PELEM_TEXT_IND,
    PELEM_INT_IND,
    PELEM_EDGE_IND,
    PELEM_LINETYPE,
    PELEM_LINEWIDTH,
    PELEM_LINE_COLR_IND,
    PELEM_MARKER_TYPE,
    PELEM_MARKER_SIZE,
    PELEM_MARKER_COLR_IND,
    PELEM_TEXT_FONT,
    PELEM_TEXT_PREC,
    PELEM_CHAR_EXPAN,
    PELEM_CHAR_SPACE,
    PELEM_TEXT_COLR_IND,
    PELEM_CHAR_HT,
    PELEM_CHAR_UP_VEC,
    PELEM_TEXT_PATH,
    PELEM_TEXT_ALIGN,
    PELEM_ANNO_CHAR_HT,
    PELEM_ANNO_CHAR_UP_VEC,
    PELEM_ANNO_PATH,
    PELEM_ANNO_ALIGN,
    PELEM_ANNO_STYLE,
    PELEM_INT_STYLE,
    PELEM_INT_STYLE_IND,
    PELEM_INT_COLR_IND,
    PELEM_EDGE_FLAG,
    PELEM_EDGETYPE,
    PELEM_EDGEWIDTH,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    PELEM_EDGE_COLR_IND,
    PELEM_PAT_SIZE,
    PELEM_PAT_REF_POINT_VECS,
    PELEM_PAT_REF_POINT,
    PELEM_ADD_NAMES_SET,
    PELEM_REMOVE_NAMES_SET,
    PELEM_INDIV_ASF,
    PELEM_HLHSR_ID,
    PELEM_LOCAL_MODEL_TRAN3,
    PELEM_LOCAL_MODEL_TRAN,
    PELEM_GLOBAL_MODEL_TRAN3,
    PELEM_GLOBAL_MODEL_TRAN,
    PELEM_MODEL_CLIP_VOL3,
    PELEM_MODEL_CLIP_VOL,
    PELEM_MODEL_CLIP_IND,
    PELEM_RESTORE_MODEL_CLIP_VOL,
    PELEM_VIEW_IND,
    PELEM_EXEC_STRUCT,
    PELEM_LABEL,
    PELEM_APPL_DATA,
    PELEM_GSE,
    PELEM_PICK_ID
} Pelem_type;
```

Pelem_type_list ELEMENT TYPE LIST

```
typedef struct {
  Pint        num_elem_types;     /* number of element types */
  Pelem_type  *elem_types;        /* list of element types   */
} Pelem_type_list;
```

Perr_mode ERROR MODE

```
typedef enum {
    PERR_OFF,
    PERR_ON
} Perr_mode;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pfilter FILTER

```
typedef struct {
  Pint_list  incl_set;      /* inclusion set */
  Pint_list  excl_set;      /* exclusion set */
} Pfilter;
```

Pfilter_list FILTER LIST

```
typedef struct {
  Pint     num_filters;     /* number of filters */
  Pfilter  *filters;        /* list of filters   */
} Pfilter_list;
```

Pfloat_size FLOAT SIZE

```
typedef struct {
  Pfloat  size_x;      /* x size */
  Pfloat  size_y;      /* y size */
} Pfloat_size;
```

Pfloat_size3 FLOAT SIZE 3

```
typedef struct {
  Pfloat  size_x;      /* x size */
  Pfloat  size_y;      /* y size */
  Pfloat  size_z;      /* z size */
} Pfloat_size3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pgse_id_dep GSE IDENTIFIER DEPENDENCY

```
typedef struct {
  Pint         id;      /* GSE identifier                                */
  Pws_dep_ind  ind;     /* workstation independent/dependent indicator */
} Pgse_id_dep;
```

Pgse_id_dep_list GSE IDENTIFIER DEPENDENCY LIST

```
typedef struct {
  Pint         num_id_facs;     /* number of identifiers/dependency element */
  Pgse_id_dep  *id_facs;        /* list of GSE facilities                   */
} Pgse_id_dep_list;
```

Phalf_space HALF SPACE

```
typedef struct {
  Ppoint  point;     /* point  */
  Pvec    norm;      /* normal */
} Phalf_space;
```

Phalf_space3 HALF SPACE 3

```
typedef struct {
  Ppoint3  point;     /* point  */
  Pvec3    norm;      /* normal */
} Phalf_space3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Phalf_space_list HALF SPACE LIST

```
typedef struct {
  Pint         num_half_spaces;     /* number of half spaces */
  Phalf_space  *half_spaces;        /* list of half spaces   */
} Phalf_space_list;
```

Phalf_space_list3 HALF SPACE LIST 3

```
typedef struct {
  Pint          num_half_spaces;     /* number of half spaces */
  Phalf_space3  *half_spaces;        /* list of half spaces   */
} Phalf_space_list3;
```

Phls HUE LIGHTNESS SATURATION

```
typedef struct {
  Pfloat  hue;           /* hue        */
  Pfloat  lightness;     /* lightness  */
  Pfloat  satur;         /* saturation */
} Phls;
```

Phor_text_align HORIZONTAL TEXT ALIGNMENT

```
typedef enum {
    PHOR_NORM,
    PHOR_LEFT,
    PHOR_CTR,
    PHOR_RIGHT
} Phor_text_align;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Phsv HUE SATURATION VALUE

```
typedef struct {
  Pfloat  hue;        /* hue        */
  Pfloat  satur;      /* saturation */
  Pfloat  value;      /* value      */
} Phsv;
```

Pinq_type INQUIRE TYPE

```
typedef enum {
   PINQ_SET,
   PINQ_REALIZED
} Pinq_type;
```

Pint_attrs INTERIOR ATTRIBUTES

```
typedef struct {
  Pasf         style_asf;         /* interior style asf        */
  Pasf         style_ind_asf;     /* interior style index asf  */
  Pasf         colr_ind_asf;      /* interior colour index asf */
  Pint         ind;               /* interior index            */
  Pint_bundle  bundle;            /* interior bundle           */
} Pint_attrs;
```

Pint_bundle INTERIOR BUNDLE

```
typedef struct {
  Pint_style  style;         /* interior style        */
  Pint        style_ind;     /* interior style index  */
  Pint        colr_ind;      /* interior colour index */
} Pint_bundle;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pint_facs INTERIOR FACILITIES

```
typedef struct {
  Pint        num_int_styles;     /* number of interior styles          */
  Pint_style  int_styles[5];      /* list of available interior styles  */
  Pint_list   hatch_styles;       /* list of available hatch styles     */
  Pint        num_pred_inds;      /* number of predefined bundle indices */
} Pint_facs;
```

Pint_list INTEGER LIST

```
typedef struct {
  Pint  num_ints;      /* number of Pints in list */
  Pint  *ints;         /* list of integers        */
} Pint_list;
```

Pint_size INTEGER SIZE

```
typedef struct {
  Pint  size_x;      /* x size */
  Pint  size_y;      /* y size */
} Pint_size;
```

Pint_size3 INTEGER SIZE 3

```
typedef struct {
  Pint  size_x;      /* x size */
  Pint  size_y;      /* y size */
  Pint  size_z;      /* z size */
} Pint_size3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pint_style INTERIOR STYLE

```
typedef enum {
    PSTYLE_HOLLOW,
    PSTYLE_SOLID,
    PSTYLE_PAT,
    PSTYLE_HATCH,
    PSTYLE_EMPTY
} Pint_style;
```

Pin_class INPUT CLASS

```
typedef enum {
    PIN_NONE,
    PIN_LOC,
    PIN_STROKE,
    PIN_VAL,
    PIN_CHOICE,
    PIN_PICK,
    PIN_STRING
} Pin_class;
```

Pin_status INPUT STATUS

```
typedef enum {
    PIN_STATUS_NONE,
    PIN_STATUS_OK,
    PIN_STATUS_NO_IN
} Pin_status;
```

Plimit LIMIT

```
typedef struct {
  Pfloat  x_min;      /* x min */
  Pfloat  x_max;      /* x max */
  Pfloat  y_min;      /* y min */
  Pfloat  y_max;      /* y max */
} Plimit;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Plimit3 LIMIT 3

```
typedef struct {
  Pfloat  x_min;      /* x min */
  Pfloat  x_max;      /* x max */
  Pfloat  y_min;      /* y min */
  Pfloat  y_max;      /* y max */
  Pfloat  z_min;      /* z min */
  Pfloat  z_max;      /* z max */
} Plimit3;
```

Pline_attrs POLYLINE ATTRIBUTES

```
typedef struct {
  Pasf          type_asf;         /* linetype asf           */
  Pasf          width_asf;        /* linewidth asf          */
  Pasf          colr_ind_asf;     /* line colour index asf */
  Pint          ind;              /* line index             */
  Pline_bundle  bundle;           /* line bundle            */
} Pline_attrs;
```

Pline_bundle POLYLINE BUNDLE

```
typedef struct {
  Pint    type;          /* linetype                */
  Pfloat  width;         /* linewidth scale factor */
  Pint    colr_ind;      /* colour index            */
} Pline_bundle;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pline_facs POLYLINE FACILITIES

```
typedef struct {
  Pint_list  types;            /* list of linetypes                    */
  Pint       num_widths;       /* number of available linewidths       */
  Pfloat     nom_width;        /* nominal linewidth                    */
  Pfloat     min_width;        /* min linewidth                        */
  Pfloat     max_width;        /* max linewidth                        */
  Pint       num_pred_inds;    /* number of predefined bundle indices */
} Pline_facs;
```

Pline_fill_ctrl_flag POLYLINE FILL CONTROL FLAG

```
typedef enum {
    PFLAG_LINE,
    PFLAG_FILL,
    PFLAG_FILL_SET
} Pline_fill_ctrl_flag;
```

Pmarker_attrs POLYMARKER ATTRIBUTES

```
typedef struct {
  Pasf           type_asf;       /* marker type asf              */
  Pasf           size_asf;       /* marker size scale factor asf */
  Pasf           colr_ind_asf;   /* marker colour index asf      */
  Pint           ind;            /* marker index                 */
  Pmarker_bundle bundle;         /* marker bundle                */
} Pmarker_attrs;
```

Pmarker_bundle POLYMARKER BUNDLE

```
typedef struct {
  Pint    type;          /* marker type              */
  Pfloat  size;          /* marker size scale factor */
  Pint    colr_ind;      /* colour index             */
} Pmarker_bundle;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Pmarker_facs POLYMARKER FACILITIES

```
typedef struct {
  Pint_list  types;            /* list of marker types                */
  Pint       num_sizes;        /* number of available marker sizes    */
  Pfloat     nom_size;         /* nominal marker size                 */
  Pfloat     min_size;         /* min marker size                     */
  Pfloat     max_size;         /* max marker size                     */
  Pint       num_pred_inds;    /* number of predefined bundle indices */
} Pmarker_facs;
```

Pmatrix MATRIX

```
typedef Pfloat Pmatrix[3][3];
```

Pmatrix3 MATRIX 3

```
typedef Pfloat Pmatrix3[4][4];
```

Pmod_mode MODIFICATION MODE

```
typedef enum {
    PMODE_NIVE,
    PMODE_UWOR,
    PMODE_UQUM
} Pmod_mode;
```

Pmore_simult_events MORE SIMULTANEOUS EVENTS

```
typedef enum {
    PSIMULT_NO_MORE,
    PSIMULT_MORE
} Pmore_simult_events;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pnum_in NUMBER OF INPUT DEVICES

```
typedef struct {
  Pint  loc;        /* locators  */
  Pint  stroke;     /* strokes   */
  Pint  val;        /* valuators */
  Pint  choice;     /* choices   */
  Pint  pick;       /* picks     */
  Pint  string;     /* strings   */
} Pnum_in;
```

Popen_struct_status OPEN STRUCTURE STATUS

```
typedef enum {
    PSTRUCT_NONE,
    PSTRUCT_OPEN
} Popen_struct_status;
```

Pop_mode OPERATING MODE

```
typedef enum {
    POP_REQ,
    POP_SAMPLE,
    POP_EVENT
} Pop_mode;
```

Pparal PARALLELOGRAM

```
typedef struct {
  Ppoint3  p;     /* point p */
  Ppoint3  q;     /* point q */
  Ppoint3  r;     /* point r */
} Pparal;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Ppath_order PATH ORDER

```
typedef enum {
    PORDER_TOP_FIRST,
    PORDER_BOTTOM_FIRST
} Ppath_order;
```

Ppat_rep PATTERN REPRESENTATION

```
typedef struct {
  Pint_size  dims;          /* pattern's dimensions */
  Pint       *colr_array;   /* colour index array   */
} Ppat_rep;
```

Ppick_path PICK PATH

```
typedef struct {
  Pint             depth;       /* pick path depth */
  Ppick_path_elem  *path_list;  /* pick path list  */
} Ppick_path;
```

Ppick_path_elem PICK PATH ELEMENT

```
typedef struct {
  Pint  struct_id;     /* structure identifier */
  Pint  pick_id;       /* pick identifier      */
  Pint  elem_pos;      /* element position     */
} Ppick_path_elem;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Ppoint POINT

```
typedef struct {
  Pfloat  x;      /* x coordinate */
  Pfloat  y;      /* y coordinate */
} Ppoint;
```

Ppoint3 POINT 3

```
typedef struct {
  Pfloat  x;      /* x coordinate */
  Pfloat  y;      /* y coordinate */
  Pfloat  z;      /* z coordinate */
} Ppoint3;
```

Ppoint_list POINT LIST

```
typedef struct {
  Pint    num_points;     /* number of Ppoints in the list */
  Ppoint  *points;        /* list of points                */
} Ppoint_list;
```

Ppoint_list3 POINT LIST 3

```
typedef struct {
  Pint     num_points;     /* number of Ppoint3s in the list */
  Ppoint3  *points;        /* list of points                 */
} Ppoint_list3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Ppoint_list_list POINT LIST LIST

```
typedef struct {
  Pint         num_point_lists;     /* number of point lists */
  Ppoint_list  *point_lists;        /* list of point lists   */
} Ppoint_list_list;
```

Ppoint_list_list3 POINT LIST LIST 3

```
typedef struct {
  Pint          num_point_lists;     /* number of point lists */
  Ppoint_list3  *point_lists;        /* list of point lists   */
} Ppoint_list_list3;
```

Pposted_struct POSTED STRUCTURE

```
typedef struct {
  Pint    id;           /* structure id     */
  Pfloat  disp_pri;     /* display priority */
} Pposted_struct;
```

Pposted_struct_list POSTED STRUCTURE LIST

```
typedef struct {
  Pint            num_postings;     /* number of structure postings */
  Pposted_struct  *postings;        /* list of postings             */
} Pposted_struct_list;
```

Pproj_type PROJECTION TYPE

```
typedef enum {
   PTYPE_PARAL,
   PTYPE_PERSPECT
} Pproj_type;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Ppr_switch PROMPT SWITCH

```
typedef enum {
    PPR_OFF,
    PPR_ON
} Ppr_switch;
```

Prect RECTANGLE

```
typedef struct {
  Ppoint  p;      /* point p */
  Ppoint  q;      /* point q */
} Prect;
```

Pref_flag REFERENCE FLAG

```
typedef enum {
    PFLAG_DEL,
    PFLAG_KEEP
} Pref_flag;
```

Pregen_flag REGENERATION FLAG

```
typedef enum {
    PFLAG_POSTPONE,
    PFLAG_PERFORM
} Pregen_flag;
```

Prel_pri RELATIVE PRIORITY

```
typedef enum {
    PPRI_HIGHER,
    PPRI_LOWER
} Prel_pri;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Prgb RED GREEN BLUE

```
typedef struct {
  Pfloat  red;        /* red intensity   */
  Pfloat  green;      /* green intensity */
  Pfloat  blue;       /* blue intensity  */
} Prgb;
```

Psearch_dir SEARCH DIRECTION

```
typedef enum {
    PDIR_BACKWARD,
    PDIR_FORWARD
} Psearch_dir;
```

Psearch_status SEARCH STATUS

```
typedef enum {
    PSEARCH_STATUS_FAILURE,
    PSEARCH_STATUS_SUCCESS
} Psearch_status;
```

Pstore STORE

```
typedef void *Pstore;
```

Pstruct_net_source STRUCTURE NETWORK SOURCE

```
typedef enum {
    PNET_CSS,
    PNET_AR
} Pstruct_net_source;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pstruct_st STRUCTURE STATE

```
typedef enum {
    PSTRUCT_ST_STCL,
    PSTRUCT_ST_STOP
} Pstruct_st;
```

Pstruct_status STRUCTURE STATUS

```
typedef enum {
    PSTRUCT_STATUS_NON_EXISTENT,
    PSTRUCT_STATUS_EMPTY,
    PSTRUCT_STATUS_NOT_EMPTY
} Pstruct_status;
```

Psys_st SYSTEM STATE

```
typedef enum {
    PSYS_ST_PHCL,
    PSYS_ST_PHOP
} Psys_st;
```

Ptext_align TEXT ALIGNMENT

```
typedef struct {
  Phor_text_align   hor;      /* horizontal component */
  Pvert_text_align  vert;     /* vertical component   */
} Ptext_align;
```

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577401 197 ■

Ptext_bundle TEXT BUNDLE

```
typedef struct {
  Pint        font;           /* text font                  */
  Ptext_prec  prec;           /* text precision             */
  Pfloat      char_expan;     /* character expansion factor */
  Pfloat      char_space;     /* character spacing          */
  Pint        colr_ind;       /* text colour index          */
} Ptext_bundle;
```

Ptext_facs TEXT FACILITIES

```
typedef struct {
  Pint             num_font_precs;      /* number of fonts and precisions        */
  Ptext_font_prec  *font_precs;         /* list of fonts and precisions          */
  Pint             num_char_hts;        /* number of character heights           */
  Pfloat           min_char_ht;         /* minimum height                        */
  Pfloat           max_char_ht;         /* maximum height                        */
  Pint             num_char_expans;     /* number of character expansion factors */
  Pfloat           min_char_expan;      /* minimum expansion factor              */
  Pfloat           max_char_expan;      /* maximum expansion factor              */
  Pint             num_pred_inds;       /* number of predefined bundle indices   */
} Ptext_facs;
```

Ptext_font_prec TEXT FONT AND PRECISION

```
typedef struct {
  Pint        font;     /* text font      */
  Ptext_prec  prec;     /* text precision */
} Ptext_font_prec;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Ptext_path TEXT PATH

```
typedef enum {
    PPATH_RIGHT,
    PPATH_LEFT,
    PPATH_UP,
    PPATH_DOWN
} Ptext_path;
```

Ptext_prec TEXT PRECISION

```
typedef enum {
    PPREC_STRING,
    PPREC_CHAR,
    PPREC_STROKE
} Ptext_prec;
```

Pupd_st UPDATE STATE

```
typedef enum {
    PUPD_NOT_PEND,
    PUPD_PEND
} Pupd_st;
```

Pvec VECTOR

```
typedef struct {
  Pfloat  delta_x;      /* delta x value */
  Pfloat  delta_y;      /* delta y value */
} Pvec;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pvec3 VECTOR 3

```
typedef struct {
  Pfloat  delta_x;      /* delta x value */
  Pfloat  delta_y;      /* delta y value */
  Pfloat  delta_z;      /* delta z value */
} Pvec3;
```

Pvert_text_align VERTICAL TEXT ALIGNMENT

```
typedef enum {
    PVERT_NORM,
    PVERT_TOP,
    PVERT_CAP,
    PVERT_HALF,
    PVERT_BASE,
    PVERT_BOTTOM
} Pvert_text_align;
```

Pview_map VIEW MAPPING

```
typedef struct {
  Plimit  win;          /* window limits              */
  Plimit  proj_vp;      /* projection viewport limits */
} Pview_map;
```

Pview_map3 VIEW MAPPING 3

```
typedef struct {
  Plimit      win;              /* window limits              */
  Plimit3     proj_vp;          /* projection viewport limits */
  Pproj_type  proj_type;        /* projection type            */
  Ppoint3     proj_ref_point;   /* projection reference point */
  Pfloat      view_plane;       /* view plane distance        */
  Pfloat      back_plane;       /* back plane distance        */
  Pfloat      front_plane;      /* front plane distance       */
} Pview_map3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Pview_rep VIEW REPRESENTATION

```
typedef struct {
  Pmatrix    ori_matrix;     /* orientation matrix     */
  Pmatrix    map_matrix;     /* mapping matrix         */
  Plimit     clip_limit;     /* clipping limits        */
  Pclip_ind  xy_clip;        /* X-Y clipping indicator */
} Pview_rep;
```

Pview_rep3 VIEW REPRESENTATION 3

```
typedef struct {
  Pmatrix3   ori_matrix;     /* orientation matrix       */
  Pmatrix3   map_matrix;     /* mapping matrix           */
  Plimit3    clip_limit;     /* clipping limits          */
  Pclip_ind  xy_clip;        /* X-Y clipping indicator   */
  Pclip_ind  back_clip;      /* back clipping indicator  */
  Pclip_ind  front_clip;     /* front clipping indicator */
} Pview_rep3;
```

Pvisual_st VISUAL STATE

```
typedef enum {
    PVISUAL_ST_CORRECT,
    PVISUAL_ST_DEFER,
    PVISUAL_ST_SIMULATED
} Pvisual_st;
```

Pws_cat WORKSTATION CATEGORY

```
typedef enum {
    PCAT_OUT,
    PCAT_IN,
    PCAT_OUTIN,
    PCAT_MO,
    PCAT_MI
} Pws_cat;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

Pws_class WORKSTATION CLASS

```
typedef enum {
    PCLASS_VEC,
    PCLASS_RASTER,
    PCLASS_OTHER
} Pws_class;
```

Pws_dep_ind WORKSTATION DEPENDENCY INDICATOR

```
typedef enum {
    PWS_INDEP,
    PWS_DEP
} Pws_dep_ind;
```

Pws_st WORKSTATION STATE

```
typedef enum {
    PWS_ST_WSCL,
    PWS_ST_WSOP
} Pws_st;
```

Pws_st_tables LENGTH OF WORKSTATION STATE TABLES

```
typedef struct {
  Pint  line_bundles;     /* max. # of polyline table entries   */
  Pint  mark_bundles;     /* max. # of polymarker table entries */
  Pint  text_bundles;     /* max. # of text table entries       */
  Pint  int_bundles;      /* max. # of interior table entries   */
  Pint  edge_bundles;     /* max. # of edge table entries       */
  Pint  pat_reps;         /* max. # of pattern table entries    */
  Pint  colr_reps;        /* max. # of colour table entries     */
  Pint  view_reps;        /* max. # of view table entries       */
} Pws_st_tables;
```

NOTES

1 The order of the enumerations in the types `Pclip_ind`, `Pcolr_avail`, `Pecho_switch`, and `Pdisp_surf_empty` have been reversed with respect to ISO/IEC 9592-1. This was done in order to keep the numerical value of the enumerations compatible with ISO/IEC 9593-1 (PHIGS Fortran).

2 The enumerations for the status value returned by the REQUEST input functions have been bound as one enumeration in type `Pin_status`; the values NO PICK and NO CHOICE have been bound as `PIN_STATUS_NO_IN`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

3 ISO/IEC 9592-1 has two large enumerations which include all the element types. These enumerations have been bound as one enumeration in the type `Pelem_type`. The bound value `PELEM_ALL` cannot be returned as a value by the functions INQUIRE CURRENT ELEMENT TYPE AND SIZE and INQUIRE ELEMENT TYPE AND SIZE.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6 Macro definitions

6.1 Function identifiers

The error functions require a unique mapping of the PHIGS functions to a set of numbers. The names for these function identifiers are the same as the bound PHIGS function names except that the sentinel character has been replaced by "Pfn_".

Function Name Macro	Function Number
#define Pfn_open_phigs	(0)
#define Pfn_close_phigs	(1)
#define Pfn_open_ws	(2)
#define Pfn_close_ws	(3)
#define Pfn_redraw_all_structs	(4)
#define Pfn_upd_ws	(5)
#define Pfn_set_disp_upd_st	(6)
#define Pfn_message	(7)
#define Pfn_polyline3	(8)
#define Pfn_polyline	(9)
#define Pfn_polymarker3	(10)
#define Pfn_polymarker	(11)
#define Pfn_text3	(12)
#define Pfn_text	(13)
#define Pfn_anno_text_rel3	(14)
#define Pfn_anno_text_rel	(15)
#define Pfn_fill_area3	(16)
#define Pfn_fill_area	(17)
#define Pfn_fill_area_set3	(18)
#define Pfn_fill_area_set	(19)
#define Pfn_cell_array3	(20)
#define Pfn_cell_array	(21)
#define Pfn_gdp3	(22)
#define Pfn_gdp	(23)
#define Pfn_set_line_ind	(24)
#define Pfn_set_marker_ind	(25)
#define Pfn_set_text_ind	(26)
#define Pfn_set_int_ind	(27)
#define Pfn_set_edge_ind	(28)
#define Pfn_set_linetype	(29)
#define Pfn_set_linewidth	(30)
#define Pfn_set_line_colr_ind	(31)
#define Pfn_set_marker_type	(32)
#define Pfn_set_marker_size	(33)
#define Pfn_set_marker_colr_ind	(34)
#define Pfn_set_text_font	(35)
#define Pfn_set_text_prec	(36)
#define Pfn_set_char_expan	(37)
#define Pfn_set_char_space	(38)
#define Pfn_set_text_colr_ind	(39)
#define Pfn_set_char_ht	(40)
#define Pfn_set_char_up_vec	(41)
#define Pfn_set_text_path	(42)
#define Pfn_set_text_align	(43)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define  Pfn_set_anno_char_ht                (44)
#define  Pfn_set_anno_char_up_vec            (45)
#define  Pfn_set_anno_path                   (46)
#define  Pfn_set_anno_align                  (47)
#define  Pfn_set_anno_style                  (48)
#define  Pfn_set_int_style                   (49)
#define  Pfn_set_int_style_ind               (50)
#define  Pfn_set_int_colr_ind                (51)
#define  Pfn_set_edge_flag                   (52)
#define  Pfn_set_edgetype                    (53)
#define  Pfn_set_edgewidth                   (54)
#define  Pfn_set_edge_colr_ind               (55)
#define  Pfn_set_pat_size                    (56)
#define  Pfn_set_pat_ref_point_vecs          (57)
#define  Pfn_set_pat_ref_point               (58)
#define  Pfn_add_names_set                   (59)
#define  Pfn_remove_names_set                (60)
#define  Pfn_set_indiv_asf                   (61)
#define  Pfn_set_line_rep                    (62)
#define  Pfn_set_marker_rep                  (63)
#define  Pfn_set_text_rep                    (64)
#define  Pfn_set_int_rep                     (65)
#define  Pfn_set_edge_rep                    (66)
#define  Pfn_set_pat_rep                     (67)
#define  Pfn_set_colr_rep                    (68)
#define  Pfn_set_highl_filter                (69)
#define  Pfn_set_invis_filter                (70)
#define  Pfn_set_colr_model                  (71)
#define  Pfn_set_hlhsr_id                    (72)
#define  Pfn_set_hlhsr_mode                  (73)
#define  Pfn_set_local_tran3                 (74)
#define  Pfn_set_local_tran                  (75)
#define  Pfn_set_global_tran3                (76)
#define  Pfn_set_global_tran                 (77)
#define  Pfn_set_model_clip_vol3             (78)
#define  Pfn_set_model_clip_vol              (79)
#define  Pfn_set_model_clip_ind              (80)
#define  Pfn_restore_model_clip_vol          (81)
#define  Pfn_set_view_ind                    (82)
#define  Pfn_set_view_rep3                   (83)
#define  Pfn_set_view_rep                    (84)
#define  Pfn_set_view_tran_in_pri            (85)
#define  Pfn_set_ws_win3                     (86)
#define  Pfn_set_ws_win                      (87)
#define  Pfn_set_ws_vp3                      (88)
#define  Pfn_set_ws_vp                       (89)
#define  Pfn_open_struct                     (90)
#define  Pfn_close_struct                    (91)
#define  Pfn_exec_struct                     (92)
#define  Pfn_label                           (93)
#define  Pfn_appl_data                       (94)
#define  Pfn_gse                             (95)
#define  Pfn_set_edit_mode                   (96)
#define  Pfn_copy_all_elems_struct           (97)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
#define  Pfn_set_elem_ptr                (98)
#define  Pfn_offset_elem_ptr             (99)
#define  Pfn_set_elem_ptr_label          (100)
#define  Pfn_del_elem                    (101)
#define  Pfn_del_elem_range              (102)
#define  Pfn_del_elems_labels            (103)
#define  Pfn_empty_struct                (104)
#define  Pfn_del_struct                  (105)
#define  Pfn_del_struct_net              (106)
#define  Pfn_del_all_structs             (107)
#define  Pfn_change_struct_id            (108)
#define  Pfn_change_struct_refs          (109)
#define  Pfn_change_struct_id_refs       (110)
#define  Pfn_post_struct                 (111)
#define  Pfn_unpost_struct               (112)
#define  Pfn_unpost_all_structs          (113)
#define  Pfn_open_ar_file                (114)
#define  Pfn_close_ar_file               (115)
#define  Pfn_ar_structs                  (116)
#define  Pfn_ar_struct_nets              (117)
#define  Pfn_ar_all_structs              (118)
#define  Pfn_set_conf_res                (119)
#define  Pfn_ret_struct_ids              (120)
#define  Pfn_ret_structs                 (121)
#define  Pfn_ret_struct_nets             (122)
#define  Pfn_ret_all_structs             (123)
#define  Pfn_ret_paths_ances             (124)
#define  Pfn_ret_paths_descs             (125)
#define  Pfn_del_structs_ar              (126)
#define  Pfn_del_struct_nets_ar          (127)
#define  Pfn_del_all_structs_ar          (128)
#define  Pfn_set_pick_id                 (129)
#define  Pfn_set_pick_filter             (130)
#define  Pfn_init_loc3                   (131)
#define  Pfn_init_loc                    (132)
#define  Pfn_init_stroke3                (133)
#define  Pfn_init_stroke                 (134)
#define  Pfn_init_val3                   (135)
#define  Pfn_init_val                    (136)
#define  Pfn_init_choice3                (137)
#define  Pfn_init_choice                 (138)
#define  Pfn_init_pick3                  (139)
#define  Pfn_init_pick                   (140)
#define  Pfn_init_string3                (141)
#define  Pfn_init_string                 (142)
#define  Pfn_set_loc_mode                (143)
#define  Pfn_set_stroke_mode             (144)
#define  Pfn_set_val_mode                (145)
#define  Pfn_set_choice_mode             (146)
#define  Pfn_set_pick_mode               (147)
#define  Pfn_set_string_mode             (148)
#define  Pfn_req_loc3                    (149)
#define  Pfn_req_loc                     (150)
#define  Pfn_req_stroke3                 (151)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define  Pfn_req_stroke                 (152)
#define  Pfn_req_val                    (153)
#define  Pfn_req_choice                 (154)
#define  Pfn_req_pick                   (155)
#define  Pfn_req_string                 (156)
#define  Pfn_sample_loc3                (157)
#define  Pfn_sample_loc                 (158)
#define  Pfn_sample_stroke3             (159)
#define  Pfn_sample_stroke              (160)
#define  Pfn_sample_val                 (161)
#define  Pfn_sample_choice              (162)
#define  Pfn_sample_pick                (163)
#define  Pfn_sample_string              (164)
#define  Pfn_await_event                (165)
#define  Pfn_flush_events               (166)
#define  Pfn_get_loc3                   (167)
#define  Pfn_get_loc                    (168)
#define  Pfn_get_stroke3                (169)
#define  Pfn_get_stroke                 (170)
#define  Pfn_get_val                    (171)
#define  Pfn_get_choice                 (172)
#define  Pfn_get_pick                   (173)
#define  Pfn_get_string                 (174)
#define  Pfn_write_item                 (175)
#define  Pfn_get_item_type              (176)
#define  Pfn_read_item                  (177)
#define  Pfn_interpret_item             (178)
#define  Pfn_set_err_hand_mode          (179)
#define  Pfn_escape                     (180)
#define  Pfn_set_err_hand               (181)
```

6.2 Error codes

```
/*                              <0          Implementation Dependent Errors */

#define  PE_NO_ERROR            (0)     /* No Error */

/* State Errors */
#define  PE_NOT_PHCL            (1)     /* Ignoring function, function requires
                                           state (PHCL,WSCL,STCL,ARCL) */
#define  PE_NOT_PHOP            (2)     /* Ignoring function, function requires
                                           state (PHOP,*,*,*) */
#define  PE_NOT_WSOP            (3)     /* Ignoring function, function requires
                                           state (PHOP,WSOP,*,*) */
#define  PE_NOT_CL              (4)     /* Ignoring function, function requires
                                           state (PHOP,WSCL,STCL,ARCL) */
#define  PE_NOT_STOP            (5)     /* Ignoring function, function requires
                                           state (PHOP,*,STOP,*) */
#define  PE_NOT_STCL            (6)     /* Ignoring function, function requires
                                           state (PHOP,*,STCL,*) */
#define  PE_NOT_AROP            (7)     /* Ignoring function, function requires
                                           state (PHOP,*,*,AROP) */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* Workstation Errors */
#define PE_BAD_CONN_ID          (50)    /* Ignoring function, connection iden-
                                           tifier not recognized by the imple-
                                           mentation */
#define PE_WS_TYPE              (51)    /* Ignoring function, this information
                                           is not yet available for this works-
                                           tation type; open a workstation of
                                           this type and use the specific works-
                                           tation type */
#define PE_BAD_WS_TYPE          (52)    /* Ignoring function, workstation type
                                           not recognized by the implementation
                                           */
#define PE_DUP_WS_ID            (53)    /* Ignoring function, workstation iden-
                                           tifier already is in use */
#define PE_WS_NOT_OPEN          (54)    /* Ignoring function, the specified
                                           workstation is not open */
#define PE_NO_OPEN_WS           (55)    /* Ignoring function, workstation cannot
                                           be opened for an implementation
                                           dependent reason */
#define PE_WS_NOT_MO            (56)    /* Ignoring function, specified worksta-
                                           tion is not of category MO */
#define PE_WS_MI                (57)    /* Ignoring function, specified worksta-
                                           tion is of category MI */
#define PE_WS_NOT_MI            (58)    /* Ignoring function, specified worksta-
                                           tion is not of category MI */
#define PE_WS_NO_OUTPUT         (59)    /* Ignoring function, the specified
                                           workstation does not have output
                                           capability (i.e., the workstation
                                           category is neither OUTPUT, OUTIN,
                                           nor MO) */
#define PE_WS_NOT_OUTIN         (60)    /* Ignoring function, specified worksta-
                                           tion is not of category OUTIN */
#define PE_WS_NO_INPUT          (61)    /* Ignoring function, specified worksta-
                                           tion is neither of category INPUT nor
                                           of category OUTIN */
#define PE_WS_NOT_OUT           (62)    /* Ignoring function, specified worksta-
                                           tion is neither of category OUTPUT
                                           nor of category OUTIN */
#define PE_MAX_WS               (63)    /* Ignoring function, opening this
                                           workstation would exceed the maximum
                                           number of simultaneously open works-
                                           tations */
#define PE_NO_GDP               (64)    /* Ignoring function, the specified
                                           workstation type is not able to gen-
                                           erate the specified generalized draw-
                                           ing primitive */

/* Output Attribute Errors */
#define PE_BUN_IND_LT_1         (100)   /* Ignoring function, the bundle index
                                           value is less than one */
#define PE_REP_UNDEF            (101)   /* Ignoring function, the specified
                                           representation has not been defined
                                           */
#define PE_REP_NOT_PREDEF       (102)   /* Ignoring function, the specified
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577412 T72 ■


```
                                        representation has not been prede-
                                        fined on this workstation */
#define PE_MAX_BUN              (103)   /* Ignoring function, setting this bun-
                                        dle table entry would exceed the max-
                                        imum number of entries allowed in the
                                        workstation bundle table */
#define PE_BAD_LINETYPE         (104)   /* Ignoring function, the specified
                                        linetype is not available on the
                                        specified workstation */
#define PE_BAD_MARKER_TYPE      (105)   /* Ignoring function, the specified
                                        marker type is not available on the
                                        specified workstation */
#define PE_BAD_FONT             (106)   /* Ignoring function, the specified font
                                        is not available for the requested
                                        text precision on the specified
                                        workstation */
#define PE_BAD_EDGETYPE         (107)   /* Ignoring function, the specified
                                        edgetype is not available on the
                                        specified workstation */
#define PE_BAD_INT_STYLE        (108)   /* Ignoring function, the specified
                                        interior style is not available on
                                        the workstation */
#define PE_NO_PAT               (109)   /* Ignoring function, interior style
                                        PATTERN is not supported on the
                                        workstation */
#define PE_BAD_COLR_MODEL       (110)   /* Ignoring function, the specified
                                        colour model is not available on the
                                        workstation */
#define PE_BAD_HLHSR_MODE       (111)   /* Ignoring function, the specified
                                        HLHSR mode is not available on the
                                        specified workstation */
#define PE_PAT_IND_LT_1         (112)   /* Ignoring function, the pattern index
                                        value is less than one */
#define PE_COLR_IND_LT_0        (113)   /* Ignoring function, the colour index
                                        value is less than zero */
#define PE_VIEW_IND_LT_0        (114)   /* Ignoring function, the view index
                                        value is less than zero */
#define PE_VIEW_IND_LT_1        (115)   /* Ignoring function, the view index
                                        value is less than one */
#define PE_BAD_PAT_DIM          (116)   /* Ignoring function, one of the dimen-
                                        sions of pattern colour array is less
                                        than one */
#define PE_BAD_COLR_DIM         (117)   /* Ignoring function, one of the dimen-
                                        sions of the colour index array is
                                        less than zero */
#define PE_BAD_COLR             (118)   /* Ignoring function, one of the com-
                                        ponents of the colour specification
                                        is out of range. The valid range is
                                        dependent upon the current colour
                                        model */

/* Transformations and Viewing Errors */
#define PE_MAX_VIEW             (150)   /* Ignoring function, setting this view
                                        table entry would exceed the maximum
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577413 909 ■


```
                                        number of entries allowed in the
                                        workstations view table */
#define PE_INVALID_WINDOW       (151)   /* Ignoring function, invalid window;
                                        XMIN >= XMAX, YMIN >= YMAX or ZMIN >
                                        ZMAX */
#define PE_INVALID_VIEWPORT     (152)   /* Ignoring function, invalid viewport;
                                        XMIN >= XMAX, YMIN >= YMAX or ZMIN >
                                        ZMAX */
#define PE_INVALID_CLIP         (153)   /* Ignoring function, invalid view clip-
                                        ping limits; XMIN >= XMAX, YMIN >=
                                        YMAX or ZMIN > ZMAX */
#define PE_BAD_CLIP             (154)   /* Ignoring function, the view clipping
                                        limits are not within NPC range */
#define PE_BAD_PROJ_VIEWPORT    (155)   /* Ignoring function, the projection
                                        viewport limits are not withing NPC
                                        range */
#define PE_BAD_WS_WINDOW        (156)   /* Ignoring function, the workstation
                                        window limits are not within NPC
                                        range */
#define PE_BAD_WS_VIEWPORT      (157)   /* Ignoring function, the workstation
                                        viewport is not within display space
                                        */
#define PE_BAD_PLANES           (158)   /* Ignoring function, front plane and
                                        back plane distances are equal when
                                        z-extent of the projection viewport
                                        is zero */
#define PE_BAD_VPN              (159)   /* Ignoring function, the view plane
                                        normal vector has length zero */
#define PE_BAD_VUP              (160)   /* Ignoring function, the view up vector
                                        has length zero */
#define PE_BAD_VUP_VPN          (161)   /* Ignoring function, the view up and
                                        view plane normal vectors are paral-
                                        lel thus the viewing coordinate sys-
                                        tem cannot be established */
#define PE_BAD_PRP              (162)   /* Ignoring function, the projection
                                        reference point is between the front
                                        and back planes */
#define PE_PRP_VIEW_PLANE       (163)   /* Ignoring function, the projection
                                        reference point cannot be positioned
                                        on the view plane */
#define PE_FRONT_BACK           (164)   /* Ignoring function, the back plane is
                                        in front of the front plane */

/* Structure Errors */
#define PE_IGNORE_STRUCTS       (200)   /* Warning, ignoring structures that do
                                        not exist */
#define PE_BAD_STRUCT           (201)   /* Ignoring function, the specified
                                        structure does not exist */
#define PE_BAD_ELEMENT          (202)   /* Ignoring function, the specified ele-
                                        ment does not exist */
#define PE_BAD_PATH             (203)   /* Ignoring function, specified starting
                                        path not found in CSS */
#define PE_BAD_CEILING_IND      (204)   /* Ignoring function, specified search
                                        ceiling index out of range */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577414 845 ■


```
#define  PE_NO_LABEL            (205)   /* Ignoring function, the label does not
                                           exist in the open structure between
                                           the element pointer and the end of
                                           the structure */
#define  PE_NO_LABELS           (206)   /* Ignoring function, one or both of the
                                           labels does not exist in the open
                                           structure between the element pointer
                                           and the end of the structure */
#define  PE_BAD_PATH_DEPTH      (207)   /* Ignoring function, the specified
                                           path depth is less than zero (0) */
#define  PE_BAD_DISP_PRI        (208)   /* Ignoring function, the display prior-
                                           ity is out of range */

/* Input Errors */
#define  PE_NO_DEVICE           (250)   /* Ignoring function, the specified dev-
                                           ice is not available on the specified
                                           workstation */
#define  PE_NOT_REQUEST         (251)   /* Ignoring function, the function
                                           requires the input device to be in
                                           REQUEST mode */
#define  PE_NOT_SAMPLE          (252)   /* Ignoring function, the function
                                           requires the input device to be in
                                           SAMPLE mode */
#define  PE_BAD_PET             (253)   /* Warning, the specified prompt/echo
                                           type is not available on the speci-
                                           fied workstation. Prompt/echo type
                                           one will be used in its place */
#define  PE_INVALID_ECHO        (254)   /* Ignoring function, invalid echo
                                           area/volume; XMIN >= XMAX, YMIN >=
                                           YMAX or ZMIN > ZMAX */
#define  PE_BAD_ECHO            (255)   /* Ignoring function, one of the echo
                                           area/volume boundary points is out-
                                           side the range of the device */
#define  PE_QUEUE_OFLOW         (256)   /* Warning, the input queue has over-
                                           flowed */
#define  PE_NO_QUEUE_OFLOW      (257)   /* Ignoring function, input queue has
                                           not overflowed */
#define  PE_OFLOW_NO_GO         (258)   /* Ignoring function, input queue has
                                           overflowed, but associated worksta-
                                           tion has been closed */
#define  PE_BAD_CLASS           (259)   /* Ignoring function, the input device
                                           class of the current input report
                                           does not match the class being
                                           requested */
#define  PE_BAD_DATA_REC        (260)   /* Ignoring function, one of the fields
                                           within the input device data record
                                           is in error */
#define  PE_INVALID_VALUE       (261)   /* Ignoring function, initial value is
                                           invalid */
#define  PE_STROKE_BUF_SIZE     (262)   /* Ignoring function, number of points
                                           in the initial stroke is greater than
                                           the buffer size */
#define  PE_STRING_BUF_SIZE     (263)   /* Ignoring function, length of the ini-
                                           tial string is greater than the
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577415 781 ■


```
                                        buffer size */

/* Metafile Errors */
#define PE_ILLEGAL_ITEM_TYPE  (300)   /* Ignoring function, item type is not
                                         allowed for user items */
#define PE_INVALID_ITEM_LENGTH        (301)/*Ignoring function, item length is
                                         invalid */
#define PE_METAFILE_EMPTY     (302)   /* Ignoring function, no item is left in
                                         metafile input */
#define PE_INVALID_ITEM       (303)   /* Ignoring function, metafile item is
                                         invalid */
#define PE_BAD_ITEM_TYPE      (304)   /* Ignoring function, item type is
                                         unknown */
#define PE_BAD_ITEM_REC       (305)   /* Ignoring function, content of item
                                         data record is invalid for the speci-
                                         fied item type */
#define PE_MAX_ITEM_LENGTH    (306)   /* Ignoring function, maximum item data
                                         record length is invalid */
#define PE_USER_ITEM          (307)   /* Ignoring function, user item cannot
                                         be interpreted */

/* Escape Errors */
#define PE_ESCAPE_NOT_AVAIL   (350)   /* Warning, the specified escape is not
                                         available on one or more workstations
                                         in this implementation. The escape
                                         will be processed by those worksta-
                                         tions on which it is available */
#define PE_BAD_ESCAPE_DATA    (351)   /* Ignoring function, one of the fields
                                         within the escape data record is in
                                         error */

/* Archival and Retrieval Errors */
#define PE_AR_CANT_OPEN       (400)   /* Ignoring function, the archive file
                                         cannot be opened */
#define PE_MAX_AR             (401)   /* Ignoring function, opening this
                                         archive file would exceed the maximum
                                         number of simultaneously open archive
                                         files */
#define PE_DUP_AR_ID          (402)   /* Ignoring function, archive file iden-
                                         tifier already in use */
#define PE_BAD_AR             (403)   /* Ignoring function, the archive file
                                         is not a PHIGS archive file */
#define PE_AR_NOT_OPEN        (404)   /* Ignoring function, the specified
                                         archive file is not open */
#define PE_NAME_CONFLICT      (405)   /* Ignoring function, name conflict
                                         occurred while conflict resolution
                                         flag has value ABANDON */
#define PE_AR_FULL            (406)   /* Warning, the archive file is full.
                                         Any structures that were archived
                                         were archived in total */
#define PE_AR_NO_STRUCT       (407)   /* Warning, some of the specified struc-
                                         tures do not exist on the archive
                                         file */
#define PE_AR_NO_STRUCT_EMPTY (408)   /* Warning, some of the specified
```


--`,,`-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                        structures do not exist on the
                                        archive file.  PHIGS will create
                                        empty structure in their places */

/* Miscellaneous Errors */
#define PE_BAD_ERROR_FILE     (450)   /* Ignoring function, the specified
                                        error file is invalid */

/* System Errors */
#define PE_OFLOW_PHIGS        (900)   /* Storage overflow has occurred in
                                        PHIGS */
#define PE_OFLOW_CSS          (901)   /* Storage overflow has occurred in CSS
                                        */
#define PE_IO_ERROR_READ      (902)   /* Input/Output error has occurred while
                                        reading */
#define PE_IO_ERROR_WRITE     (903)   /* Input/Output error has occurred while
                                        writing */
#define PE_IO_ERROR_TO_WS     (904)   /* Input/Output error has occurred while
                                        sending data to a workstation */
#define PE_IO_ERROR_FROM_WS   (905)   /* Input/Output error has occurred
                                        receiving data from a workstation */
#define PE_IO_ERROR_LIB       (906)   /* Input/Output error has occurred dur-
                                        ing program library management */
#define PE_IO_ERROR_WDT       (907)   /* Input/Output error has occurred while
                                        reading workstation description table
                                        */
#define PE_ARITHMETIC_ERROR   (908)   /* Arithmetic error has occurred */

/*      Binding Specific Errors */
#define PE_START_IND_INVAL    (2200)  /* Ignoring function, start index is out
                                        of range */
#define PE_LIST_LENGTH_LT_ZERO        (2201)/*Ignoring function, the length of
                                        the application's list is negative */
#define PE_ENUM_TYPE_INVAL    (2202)  /* Ignoring function, enumeration type
                                        out of range */
#define PE_ALLOC_STORE        (2203)  /* Ignoring function, error while allo-
                                        cating a Store */
#define PE_ALLOC_STORE_MEM    (2204)  /* Ignoring function, error while allo-
                                        cating memory for a Store */
```

Reserved error numbers: 2202-2299

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577417 554 ■

6.3 Miscellaneous

6.3.1 Linetypes

```
#define  PLINE_SOLID                    (1)
#define  PLINE_DASH                     (2)
#define  PLINE_DOT                      (3)
#define  PLINE_DASH_DOT                 (4)
```

6.3.2 Marker types

```
#define  PMARKER_DOT                    (1)
#define  PMARKER_PLUS                   (2)
#define  PMARKER_ASTERISK               (3)
#define  PMARKER_CIRCLE                 (4)
#define  PMARKER_CROSS                  (5)
```

6.3.3 Annotation styles

```
#define PANNO_STYLE_UNCONNECTED         (1)
#define PANNO_STYLE_LEAD_LINE           (2)
```

6.3.4 Colour models

```
#define  PMODEL_RGB                     (1)
#define  PMODEL_CIELUV                  (2)
#define  PMODEL_HSV                     (3)
#define  PMODEL_HLS                     (4)
```

6.3.5 Prompt and Echo Types

```
#define  PLOC_DEF                       (1)
#define  PLOC_CROSS_HAIR                (2)
#define  PLOC_TRACK_CROSS               (3)
#define  PLOC_RUB_BAND                  (4)
#define  PLOC_RECT                      (5)
#define  PLOC_DIGIT                     (6)

#define  PSTROKE_DEF                    (1)
#define  PSTROKE_DIGIT                  (2)
#define  PSTROKE_MARKER                 (3)
#define  PSTROKE_LINE                   (4)

#define  PVAL_DEF                       (1)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define  PVAL_GRAPH                    (2)
#define  PVAL_DIGIT                    (3)

#define  PCHOICE_DEF                   (1)
#define  PCHOICE_PR_ECHO               (2)
#define  PCHOICE_STRING_PR             (3)
#define  PCHOICE_STRING_IN             (4)
#define  PCHOICE_STRUCT                (5)

#define  PPICK_DEF                     (1)
#define  PPICK_GROUP_HIGHL             (2)
#define  PPICK_STRUCT_NETWORK          (3)

#define PSTRING_DEF                    (1)
```

6.3.6 Default parameters of OPEN PHIGS

```
#define  PDEF_MEM_SIZE                 ((size_t) (-1))
#define  PDEF_ERR_FILE                 ((char *) (0))
```

6.3.7 Element enumeration

```
#define  PFIRST_PHIGS_ELEM             (PELEM_POLYLINE3)
#define  PLAST_PHIGS_ELEM              (PELEM_PICK_ID)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577419 327 ■

7 C PHIGS functions

7.1 Notational conventions

The format of the binding of each PHIGS function conforms to the following template:

PHIGS Function Name

```
void   pfunction(
       Ptype0   arg0,   /* argument 0 explanation   */
       Ptype1   arg1,   /* argument 1 explanation   */
       Ptype2   arg2    /* argument 2 explanation   */
);
```

"PHIGS Function Name" is the name of the function as listed in the PHIGS standard.

The C function name bound to the PHIGS function is `pfunction`. `arg0`, `arg1`, and `arg2` are the arguments to the function and correspond to the parameters of the PHIGS function definition. `Ptype0`, `Ptype1`, and `Ptype2` are the C data types of the arguments. The definitions of these types are listed in clause 5 of this binding.

To the right of each argument declaration is a C comment field which contains a brief explanation of the argument. If the comment begins with "OUT" it means the argument is used as an output parameter; the implementation returns data to the application through this argument. Arguments without "OUT" are input parameters. The function DELETE STORE has a parameter labeled "IN/OUT"; this parameter is first used as an input paramter and then it is assigned a value by the implementation. In order to conserve space in the comment field, the symbol "#" is used as an abbreviation for "number of".

All C PHIGS functions return `void`.

7.2 Control functions

OPEN PHIGS

```
void popen_phigs(
     const char    *err_file,   /* name of error file
     size_t        mem_units    /* size_t units of memory available for buffer space
);
```

CLOSE PHIGS

```
void pclose_phigs(
     void
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

OPEN WORKSTATION

```
void popen_ws(
    Pint         ws_id,      /* workstation identifier   */
    const void   *conn_id,   /* connection identifier    */
    Pint         ws_type     /* workstation type         */
);
```

CLOSE WORKSTATION

```
void pclose_ws(
    Pint   ws_id   /* workstation identifier   */
);
```

REDRAW ALL STRUCTURES

```
void predraw_all_structs(
    Pint         ws_id,      /* workstation identifier   */
    Pctrl_flag   ctrl_flag   /* control flag             */
);
```

UPDATE WORKSTATION

```
void pupd_ws(
    Pint          ws_id,       /* workstation identifier     */
    Pregen_flag   regen_flag   /* update regeneration flag   */
);
```

SET DISPLAY UPDATE STATE

```
void pset_disp_upd_st(
    Pint          ws_id,      /* workstation identifier   */
    Pdefer_mode   def_mode,   /* deferral mode            */
    Pmod_mode     mod_mode    /* modification mode        */
);
```

MESSAGE

```
void pmessage(
    Pint          ws_id,     /* workstation identifier   */
    const char    *message   /* message string           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577421 T85 ■

7.3 Output primitive functions

POLYLINE 3

```
void ppolyline3(
    const Ppoint_list3   *point_list   /* list of points   */
);
```

POLYLINE

```
void ppolyline(
    const Ppoint_list   *point_list   /* list of points   */
);
```

POLYMARKER 3

```
void ppolymarker3(
    const Ppoint_list3   *point_list   /* list of points   */
);
```

POLYMARKER

```
void ppolymarker(
    const Ppoint_list   *point_list   /* list of points   */
);
```

TEXT 3

```
void ptext3(
    const Ppoint3   *text_pos,     /* text position           */
    const Pvec3     text_dir[2],   /* text direction vectors  */
    const char      *char_string   /* character string        */
);
```

TEXT

```
void ptext(
    const Ppoint   *text_pos,     /* text position      */
    const char     *char_string   /* character string   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ANNOTATION TEXT RELATIVE 3

```
void panno_text_rel3(
    const Ppoint3   *ref_point,    /* reference point              */
    const Pvec3     *offset,       /* annotation offset            */
    const char      *char_string   /* annotation character string  */
);
```

ANNOTATION TEXT RELATIVE

```
void panno_text_rel(
    const Ppoint   *ref_point,    /* reference point              */
    const Pvec     *offset,       /* annotation offset            */
    const char     *char_string   /* annotation character string  */
);
```

FILL AREA 3

```
void pfill_area3(
    const Ppoint_list3   *point_list   /* list of points   */
);
```

FILL AREA

```
void pfill_area(
    const Ppoint_list   *point_list   /* list of points   */
);
```

FILL AREA SET 3

```
void pfill_area_set3(
    const Ppoint_list_list3   *point_list_list   /* list of point lists   */
);
```

FILL AREA SET

```
void pfill_area_set(
    const Ppoint_list_list   *point_list_list   /* list of point lists   */
);
```

CELL ARRAY 3

```
void pcell_array3(
    const Pparal     *paral,       /* cell parallelogram   */
    const Ppat_rep   *colr_array   /* colour array         */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

CELL ARRAY

```
void pcell_array(
    const Prect      *rect,        /* cell rectangle   */
    const Ppat_rep   *colr_array   /* colour array     */
);
```

GENERALIZED DRAWING PRIMITIVE 3

```
void pgdp3(
    const Ppoint_list3   *point_list,   /* list of points           */
    Pint                 gdp3_id,       /* gdp function identifier  */
    const Pgdp_data3     *gdp_data      /* gdp data record          */
);
```

GENERALIZED DRAWING PRIMITIVE

```
void pgdp(
    const Ppoint_list   *point_list,   /* list of points           */
    Pint                gdp_id,        /* gdp function identifier  */
    const Pgdp_data     *gdp_data      /* gdp data record          */
);
```

7.4 Attribute specification functions

7.4.1 Bundled attribute selection

SET POLYLINE INDEX

```
void pset_line_ind(
    Pint   line_ind   /* polyline index   */
);
```

SET POLYMARKER INDEX

```
void pset_marker_ind(
    Pint   marker_ind   /* polymarker index   */
);
```

SET TEXT INDEX

```
void pset_text_ind(
    Pint   text_ind   /* text index   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET INTERIOR INDEX

```
void pset_int_ind(
    Pint   int_ind   /* interior index   */
);
```

SET EDGE INDEX

```
void pset_edge_ind(
    Pint   edge_ind   /* edge index   */
);
```

7.4.2 Individual attribute selection

SET LINETYPE

```
void pset_linetype(
    Pint   linetype   /* linetype   */
);
```

SET LINEWIDTH SCALE FACTOR

```
void pset_linewidth(
    Pfloat   linewidth   /* linewidth scale factor   */
);
```

SET POLYLINE COLOUR INDEX

```
void pset_line_colr_ind(
    Pint   line_colr_ind   /* polyline colour index   */
);
```

SET MARKER TYPE

```
void pset_marker_type(
    Pint   marker_type   /* marker type   */
);
```

SET MARKER SIZE SCALE FACTOR

```
void pset_marker_size(
    Pfloat   marker_size   /* marker size scale factor   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

SET POLYMARKER COLOUR INDEX

```
void pset_marker_colr_ind(
    Pint   marker_colr_ind   /* polymarker colour index   */
);
```

SET TEXT FONT

```
void pset_text_font(
    Pint   font   /* text font   */
);
```

SET TEXT PRECISION

```
void pset_text_prec(
    Ptext_prec   prec   /* text precision   */
);
```

SET CHARACTER EXPANSION FACTOR

```
void pset_char_expan(
    Pfloat   char_expan   /* character expansion factor   */
);
```

SET CHARACTER SPACING

```
void pset_char_space(
    Pfloat   char_space   /* character spacing   */
);
```

SET TEXT COLOUR INDEX

```
void pset_text_colr_ind(
    Pint   text_colr_ind   /* text colour index   */
);
```

SET CHARACTER HEIGHT

```
void pset_char_ht(
    Pfloat   char_ht   /* character height   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET CHARACTER UP VECTOR

```
void pset_char_up_vec(
    const Pvec   *char_up_vec   /* character up vector   */
);
```

SET TEXT PATH

```
void pset_text_path(
    Ptext_path   text_path   /* text path   */
);
```

SET TEXT ALIGNMENT

```
void pset_text_align(
    const Ptext_align   *text_align   /* text alignment   */
);
```

SET ANNOTATION TEXT CHARACTER HEIGHT

```
void pset_anno_char_ht(
    Pfloat   char_ht   /* character height   */
);
```

SET ANNOTATION TEXT CHARACTER UP VECTOR

```
void pset_anno_char_up_vec(
    const Pvec   *char_up_vec   /* character up vector   */
);
```

SET ANNOTATION TEXT PATH

```
void pset_anno_path(
    Ptext_path   text_path   /* text path   */
);
```

SET ANNOTATION TEXT ALIGNMENT

```
void pset_anno_align(
    const Ptext_align   *text_align   /* text alignment   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577427 4T3 ■

SET ANNOTATION STYLE

```
void pset_anno_style(
    Pint   anno_style   /* annotation style   */
);
```

SET INTERIOR STYLE

```
void pset_int_style(
    Pint_style   int_style   /* interior style   */
);
```

SET INTERIOR STYLE INDEX

```
void pset_int_style_ind(
    Pint   int_style_ind   /* interior style index   */
);
```

SET INTERIOR COLOUR INDEX

```
void pset_int_colr_ind(
    Pint   int_colr_ind   /* interior colour index   */
);
```

SET EDGE FLAG

```
void pset_edge_flag(
    Pedge_flag   edge_flag   /* edge flag   */
);
```

SET EDGETYPE

```
void pset_edgetype(
    Pint   edgetype   /* edgetype   */
);
```

SET EDGEWIDTH SCALE FACTOR

```
void pset_edgewidth(
    Pfloat   edgewidth   /* edgewidth scale factor   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET EDGE COLOUR INDEX

```
void pset_edge_colr_ind(
    Pint   edge_colr_ind   /* edge colour index   */
);
```

SET PATTERN SIZE

```
void pset_pat_size(
    const Pfloat_size   *pat_size   /* pattern size   */
);
```

SET PATTERN REFERENCE POINT AND VECTORS

```
void pset_pat_ref_point_vecs(
    const Ppoint3   *pat_ref_point,   /* pattern reference point                    */
    const Pvec3     pat_ref_vec[2]    /* reference vectors: 0 = X axis; 1 = Y axis  */
);
```

SET PATTERN REFERENCE POINT

```
void pset_pat_ref_point(
    const Ppoint   *pat_ref_point   /* pattern reference point   */
);
```

ADD NAMES TO SET

```
void padd_names_set(
    const Pint_list   *names   /* name set to be added   */
);
```

REMOVE NAMES FROM SET

```
void premove_names_set(
    const Pint_list   *names   /* name set to be removed   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

7.4.3 Aspect source flag setting

SET INDIVIDUAL ASF

```
void pset_indiv_asf(
    Paspect   asf_id,      /* attribute source identifier   */
    Pasf      asf_source   /* attribute source              */
);
```

7.4.4 Workstation attribute table definition

SET POLYLINE REPRESENTATION

```
void pset_line_rep(
    Pint                ws_id,         /* workstation identifier    */
    Pint                line_ind,      /* polyline bundle index     */
    const Pline_bundle  *line_bundle   /* polyline representation   */
);
```

SET POLYMARKER REPRESENTATION

```
void pset_marker_rep(
    Pint                  ws_id,           /* workstation identifier      */
    Pint                  marker_ind,      /* polymarker bundle index     */
    const Pmarker_bundle  *marker_bundle   /* polymarker representation   */
);
```

SET TEXT REPRESENTATION

```
void pset_text_rep(
    Pint                ws_id,         /* workstation identifier   */
    Pint                text_ind,      /* text bundle index        */
    const Ptext_bundle  *text_bundle   /* text representation      */
);
```

SET INTERIOR REPRESENTATION

```
void pset_int_rep(
    Pint                ws_id,         /* workstation identifier    */
    Pint                int_ind,       /* interior bundle index     */
    const Pint_bundle   *int_bundle    /* interior representation   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET EDGE REPRESENTATION

```
void pset_edge_rep(
    Pint                ws_id,          /* workstation identifier   */
    Pint                edge_ind,       /* edge bundle index        */
    const Pedge_bundle  *edge_bundle    /* edge representation      */
);
```

SET PATTERN REPRESENTATION

```
void pset_pat_rep(
    Pint            ws_id,          /* workstation identifier   */
    Pint            pat_ind,        /* pattern bundle index     */
    const Ppat_rep  *pat_bundle     /* pattern representation   */
);
```

SET COLOUR REPRESENTATION

```
void pset_colr_rep(
    Pint             ws_id,         /* workstation identifier   */
    Pint             colr_ind,      /* colour bundle index      */
    const Pcolr_rep  *colr_rep      /* colour representation    */
);
```

7.4.5 Workstation filter definition

SET HIGHLIGHTING FILTER

```
void pset_highl_filter(
    Pint            ws_id,      /* workstation identifier   */
    const Pfilter   *filter     /* highlighting filter      */
);
```

SET INVISIBILITY FILTER

```
void pset_invis_filter(
    Pint            ws_id,      /* workstation identifier   */
    const Pfilter   *filter     /* invisibility filter      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.4.6 Colour model control

SET COLOUR MODEL

```
void pset_colr_model(
    Pint   ws_id,       /* workstation identifier   */
    Pint   colr_model   /* colour model             */
);
```

7.4.7 HLHSR attributes

SET HLHSR IDENTIFIER

```
void pset_hlhsr_id(
    Pint   hlhsr_id   /* HLHSR identifier   */
);
```

SET HLHSR MODE

```
void pset_hlhsr_mode(
    Pint   ws_id,       /* workstation identifier   */
    Pint   hlhsr_mode   /* HLHSR mode               */
);
```

7.5 Transformation and clipping functions

7.5.1 Modelling transformations and clipping

SET LOCAL TRANSFORMATION 3

```
void pset_local_tran3(
    Pmatrix3        local_tran,    /* local transformation matrix   */
    Pcompose_type   compose_type   /* composition type              */
);
```

SET LOCAL TRANSFORMATION

```
void pset_local_tran(
    Pmatrix         local_tran,    /* local transformation matrix   */
    Pcompose_type   compose_type   /* composition type              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET GLOBAL TRANSFORMATION 3

```
void pset_global_tran3(
    Pmatrix3   global_tran   /* global transformation matrix   */
);
```

SET GLOBAL TRANSFORMATION

```
void pset_global_tran(
    Pmatrix   global_tran   /* global transformation matrix   */
);
```

SET MODELLING CLIPPING VOLUME 3

```
void pset_model_clip_vol3(
    Pint                        op,           /* operator              */
    const Phalf_space_list3   *half_spaces   /* list of half spaces   */
);
```

SET MODELLING CLIPPING VOLUME

```
void pset_model_clip_vol(
    Pint                      op,           /* operator              */
    const Phalf_space_list   *half_spaces   /* list of half spaces   */
);
```

SET MODELLING CLIPPING INDICATOR

```
void pset_model_clip_ind(
    Pclip_ind   clip_ind   /* clipping indicator   */
);
```

RESTORE MODELLING CLIPPING VOLUME

```
void prestore_model_clip_vol(
        void
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

7.5.2 View operation

SET VIEW INDEX

```
void pset_view_ind(
    Pint   view_ind   /* view index   */
);
```

SET VIEW REPRESENTATION 3

```
void pset_view_rep3(
    Pint               ws_id,      /* workstation identifier   */
    Pint               view_ind,   /* view index               */
    const Pview_rep3   *view_rep   /* view representation      */
);
```

SET VIEW REPRESENTATION

```
void pset_view_rep(
    Pint              ws_id,      /* workstation identifier   */
    Pint              view_ind,   /* view index               */
    const Pview_rep   *view_rep   /* view representation      */
);
```

SET VIEW TRANSFORMATION INPUT PRIORITY

```
void pset_view_tran_in_pri(
    Pint       ws_id,          /* workstation identifier   */
    Pint       view_ind,       /* view index               */
    Pint       ref_view_ind,   /* reference view index     */
    Prel_pri   rel_pri         /* relative priority        */
);
```

7.5.3 Workstation transformation

SET WORKSTATION WINDOW 3

```
void pset_ws_win3(
    Pint           ws_id,           /* workstation identifier     */
    const Plimit3  *ws_win_limits   /* workstation window limits  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET WORKSTATION WINDOW

```
void pset_ws_win(
    Pint          ws_id,          /* workstation identifier      */
    const Plimit  *ws_win_limits  /* workstation window limits   */
);
```

SET WORKSTATION VIEWPORT 3

```
void pset_ws_vp3(
    Pint           ws_id,          /* workstation identifier       */
    const Plimit3  *ws_vp_limits   /* workstation viewport limits  */
);
```

SET WORKSTATION VIEWPORT

```
void pset_ws_vp(
    Pint          ws_id,          /* workstation identifier       */
    const Plimit  *ws_vp_limits   /* workstation viewport limits  */
);
```

7.5.4 Utility functions to support modelling

TRANSLATE 3

```
void ptranslate3(
    const Pvec3   *trans_vec,   /* translation vector          */
    Pint          *err_ind,     /* OUT error indicator         */
    Pmatrix3      result_tran   /* OUT transformation matrix   */
);
```

TRANSLATE

```
void ptranslate(
    const Pvec   *trans_vec,   /* translation vector          */
    Pint         *err_ind,     /* OUT error indicator         */
    Pmatrix      result_tran   /* OUT transformation matrix   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577435 57T ■

SCALE 3

```
void pscale3(
    const Pvec3   *scale_vec,   /* scale factor vector          */
    Pint          *err_ind,     /* OUT error indicator          */
    Pmatrix3      result_tran   /* OUT transformation matrix    */
);
```

SCALE

```
void pscale(
    const Pvec   *scale_vec,   /* scale factor vector          */
    Pint         *err_ind,     /* OUT error indicator          */
    Pmatrix      result_tran   /* OUT transformation matrix    */
);
```

ROTATE X

```
void protate_x(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);
```

ROTATE Y

```
void protate_y(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);
```

ROTATE Z

```
void protate_z(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);
```

ROTATE

```
void protate(
    Pfloat    angle,        /* rotation angle               */
    Pint      *err_ind,     /* OUT error indicator          */
    Pmatrix   result_tran   /* OUT transformation matrix    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

COMPOSE MATRIX 3

```
void pcompose_matrix3(
    Pmatrix3   tran_a,       /* transformation matrix a   */
    Pmatrix3   tran_b,       /* transformation matrix b   */
    Pint       *err_ind,     /* OUT error indicator       */
    Pmatrix3   result_tran   /* OUT result matrix         */
);
```

COMPOSE MATRIX

```
void pcompose_matrix(
    Pmatrix    tran_a,       /* transformation matrix a   */
    Pmatrix    tran_b,       /* transformation matrix b   */
    Pint       *err_ind,     /* OUT error indicator       */
    Pmatrix    result_tran   /* OUT result matrix         */
);
```

TRANSFORM POINT 3

```
void ptran_point3(
    const Ppoint3   *point,     /* point                   */
    Pmatrix3        tran,       /* transformation matrix   */
    Pint            *err_ind,   /* OUT error indicator     */
    Ppoint3         *result     /* OUT transformed point   */
);
```

TRANSFORM POINT

```
void ptran_point(
    const Ppoint   *point,     /* point                   */
    Pmatrix        tran,       /* transformation matrix   */
    Pint           *err_ind,   /* OUT error indicator     */
    Ppoint         *result     /* OUT transformed point   */
);
```

BUILD TRANSFORMATION MATRIX 3

```
void pbuild_tran_matrix3(
    const Ppoint3   *point,       /* fixed point                */
    const Pvec3     *shift_vec,   /* shift vector               */
    Pfloat          x_angle,      /* rotation angle X           */
    Pfloat          y_angle,      /* rotation angle Y           */
    Pfloat          z_angle,      /* rotation angle Z           */
    const Pvec3     *scale_vec,   /* scale vector               */
    Pint            *err_ind,     /* OUT error indicator        */
    Pmatrix3        result_tran   /* OUT transformation matrix  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577437 342 ■

BUILD TRANSFORMATION MATRIX

```
void pbuild_tran_matrix(
    const Ppoint   *point,      /* fixed point                */
    const Pvec     *shift_vec,  /* shift vector               */
    Pfloat         angle,       /* rotation angle             */
    const Pvec     *scale_vec,  /* scale vector               */
    Pint           *err_ind,    /* OUT error indicator        */
    Pmatrix        result_tran  /* OUT transformation matrix  */
);
```

COMPOSE TRANSFORMATION MATRIX 3

```
void pcompose_tran_matrix3(
    Pmatrix3       tran,        /* transformation matrix      */
    const Ppoint3  *point,      /* fixed point                */
    const Pvec3    *shift_vec,  /* shift vector               */
    Pfloat         x_angle,     /* rotation angle X           */
    Pfloat         y_angle,     /* rotation angle Y           */
    Pfloat         z_angle,     /* rotation angle Z           */
    const Pvec3    *scale_vec,  /* scale vector               */
    Pint           *err_ind,    /* OUT error indicator        */
    Pmatrix3       result_tran  /* OUT transformation matrix  */
);
```

COMPOSE TRANSFORMATION MATRIX

```
void pcompose_tran_matrix(
    Pmatrix        tran,        /* transformation matrix      */
    const Ppoint   *point,      /* fixed point                */
    const Pvec     *shift_vec,  /* shift vector               */
    Pfloat         angle,       /* rotation angle             */
    const Pvec     *scale_vec,  /* scale vector               */
    Pint           *err_ind,    /* OUT error indicator        */
    Pmatrix        result_tran  /* OUT transformation matrix  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.5.5 Utility functions to support viewing

EVALUATE VIEW ORIENTATION MATRIX 3

```
void peval_view_ori_matrix3(
    const Ppoint3   *view_ref_point,   /* view reference point          */
    const Pvec3     *view_norm_vec,    /* view plane normal            */
    const Pvec3     *view_up_vec,      /* view up vector               */
    Pint            *err_ind,          /* OUT error indicator          */
    Pmatrix3        result_tran        /* OUT view orientation matrix  */
);
```

EVALUATE VIEW ORIENTATION MATRIX

```
void peval_view_ori_matrix(
    const Ppoint    *view_ref_point,   /* view reference point          */
    const Pvec      *view_up_vec,      /* view up vector               */
    Pint            *err_ind,          /* OUT error indicator          */
    Pmatrix         result_tran        /* OUT view orientation matrix  */
);
```

EVALUATE VIEW MAPPING MATRIX 3

```
void peval_view_map_matrix3(
    const Pview_map3   *mapping,     /* view mapping              */
    Pint               *err_ind,     /* OUT error indicator       */
    Pmatrix3           result_tran   /* OUT view mapping matrix   */
);
```

EVALUATE VIEW MAPPING MATRIX

```
void peval_view_map_matrix(
    const Pview_map   *mapping,     /* view mapping              */
    Pint              *err_ind,     /* OUT error indicator       */
    Pmatrix           result_tran   /* OUT view mapping matrix   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577439 115 ■

7.6 Structure content functions

OPEN STRUCTURE

```
void popen_struct(
    Pint   struct_id   /* structure identifier   */
);
```

CLOSE STRUCTURE

```
void pclose_struct(
     void
);
```

EXECUTE STRUCTURE

```
void pexec_struct(
    Pint   struct_id   /* structure identifier   */
);
```

LABEL

```
void plabel(
    Pint   label_id   /* label identifier   */
);
```

APPLICATION DATA

```
void pappl_data(
    const Pdata   *data   /* application data   */
);
```

GENERALIZED STRUCTURE ELEMENT

```
void pgse(
    Pint              id,         /* gse identifier    */
    const Pgse_data   *gse_data   /* gse data record   */
);
```

SET EDIT MODE

```
void pset_edit_mode(
    Pedit_mode   edit_mode   /* edit mode   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

COPY ALL ELEMENTS FROM STRUCTURE

```
void pcopy_all_elems_struct(
    Pint   struct_id   /* structure identifier   */
);
```

SET ELEMENT POINTER

```
void pset_elem_ptr(
    Pint   elem_ptr_value   /* element pointer value   */
);
```

OFFSET ELEMENT POINTER

```
void poffset_elem_ptr(
    Pint   elem_ptr_offset   /* element pointer offset   */
);
```

SET ELEMENT POINTER AT LABEL

```
void pset_elem_ptr_label(
    Pint   label_id   /* label identifier   */
);
```

DELETE ELEMENT

```
void pdel_elem(
    void
);
```

DELETE ELEMENT RANGE

```
void pdel_elem_range(
    Pint   elem_ptr1_value,   /* element pointer 1 value   */
    Pint   elem_ptr2_value    /* element pointer 2 value   */
);
```

DELETE ELEMENTS BETWEEN LABELS

```
void pdel_elems_labels(
    Pint   label1_id,   /* label 1 identifier   */
    Pint   label2_id    /* label 2 identifier   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577441 873 ■

EMPTY STRUCTURE

```
void pempty_struct(
    Pint   struct_id   /* structure id   */
);
```

7.7 Structure manipulation functions

DELETE STRUCTURE

```
void pdel_struct(
    Pint   struct_id   /* structure identifier   */
);
```

DELETE STRUCTURE NETWORK

```
void pdel_struct_net(
    Pint        struct_id,   /* structure identifier      */
    Pref_flag   ref_flag     /* reference handling flag   */
);
```

DELETE ALL STRUCTURES

```
void pdel_all_structs(
      void
);
```

CHANGE STRUCTURE IDENTIFIER

```
void pchange_struct_id(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);
```

CHANGE STRUCTURE REFERENCES

```
void pchange_struct_refs(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

CHANGE STRUCTURE IDENTIFIER AND REFERENCES

```
void pchange_struct_id_refs(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);
```

7.8 Structure display functions

POST STRUCTURE

```
void ppost_struct(
    Pint     ws_id,       /* workstation identifier   */
    Pint     struct_id,   /* structure identifier     */
    Pfloat   pri          /* priority                 */
);
```

UNPOST STRUCTURE

```
void punpost_struct(
    Pint   ws_id,      /* workstation identifier   */
    Pint   struct_id   /* structure identifier     */
);
```

UNPOST ALL STRUCTURES

```
void punpost_all_structs(
    Pint   ws_id   /* workstation identifier   */
);
```

7.9 Structure archiving functions

OPEN ARCHIVE FILE

```
void popen_ar_file(
    Pint          archive_id,     /* archive identifier   */
    const char    *archive_file   /* archive file name    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577443 646 ■

CLOSE ARCHIVE FILE

```
void pclose_ar_file(
    Pint   archive_id   /* archive identifier   */
);
```

ARCHIVE STRUCTURES

```
void par_structs(
    Pint              archive_id,   /* archive identifier              */
    const Pint_list   *struct_ids   /* list of structure identifiers   */
);
```

ARCHIVE STRUCTURE NETWORKS

```
void par_struct_nets(
    Pint              archive_id,   /* archive identifier              */
    const Pint_list   *struct_ids   /* list of structure identifiers   */
);
```

ARCHIVE ALL STRUCTURES

```
void par_all_structs(
    Pint   archive_id   /* archive identifier   */
);
```

SET CONFLICT RESOLUTION

```
void pset_conf_res(
    Pconf_res   archival_res,   /* archival conflict resolution    */
    Pconf_res   retrieval_res   /* retrieval conflict resolution   */
);
```

RETRIEVE STRUCTURE IDENTIFIERS

```
void pret_struct_ids(
    Pint        archive_id,            /* archive identifier          */
    Pint        num_elems_appl_list,   /* # elems in appl list        */
    Pint        start_ind,             /* starting index              */
    Pint_list   *ids,                  /* OUT list of structure ids   */
    Pint        *num_elems_impl_list   /* OUT # elems in impl. list   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

RETRIEVE STRUCTURES

```
void pret_structs(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);
```

RETRIEVE STRUCTURE NETWORKS

```
void pret_struct_nets(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);
```

RETRIEVE ALL STRUCTURES

```
void pret_all_structs(
    Pint   archive_id   /* archive identifier   */
);
```

RETRIEVE PATHS TO ANCESTORS

```
void pret_paths_ances(
    Pint                ar_id,       /* archive identifier       */
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);
```

If an error is detected by this function, `*paths` shall be set to NULL.

NOTE — The memory referenced by `*paths` is managed by `store`.

RETRIEVE PATHS TO DESCENDANTS

```
void pret_paths_descs(
    Pint                ar_id,       /* archive identifier       */
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);
```

If an error is detected by this function, `*paths` shall be set to NULL.

NOTE — The memory referenced by `*paths` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

---

DELETE STRUCTURES FROM ARCHIVE

```
void pdel_structs_ar(
    Pint             archive_id,   /* archive identifier             */
    const Pint_list  *struct_ids   /* list of structure identifiers  */
);
```

DELETE STRUCTURE NETWORKS FROM ARCHIVE

```
void pdel_struct_nets_ar(
    Pint             archive_id,   /* archive identifier             */
    const Pint_list  *struct_ids   /* list of structure identifiers  */
);
```

DELETE ALL STRUCTURES FROM ARCHIVE

```
void pdel_all_structs_ar(
    Pint  archive_id   /* archive identifier  */
);
```

7.10 Input functions

7.10.1 Pick identifier and filter

SET PICK IDENTIFIER

```
void pset_pick_id(
    Pint  pick_id   /* pick identifier  */
);
```

SET PICK FILTER

```
void pset_pick_filter(
    Pint           ws_id,      /* workstation identifier  */
    Pint           pick_num,   /* pick device number      */
    const Pfilter  *filter     /* pick filter             */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.10.2 Initialization of input devices

INITIALIZE LOCATOR 3

```
void pinit_loc3(
    Pint              ws_id,          /* workstation identifier     */
    Pint              loc_num,        /* locator device number      */
    Pint              init_view_ind,  /* initial view index         */
    const Ppoint3     *init_loc_pos,  /* initial locator position   */
    Pint              pet,            /* prompt and echo type       */
    const Plimit3     *echo_vol,      /* echo volume                */
    const Ploc_data3  *loc_data       /* data record                */
);
```

INITIALIZE LOCATOR

```
void pinit_loc(
    Pint              ws_id,          /* workstation identifier     */
    Pint              loc_num,        /* locator device number      */
    Pint              init_view_ind,  /* initial view index         */
    const Ppoint      *init_loc_pos,  /* initial locator position   */
    Pint              pet,            /* prompt and echo type       */
    const Plimit      *echo_area,     /* echo area                  */
    const Ploc_data   *loc_data       /* data record                */
);
```

INITIALIZE STROKE 3

```
void pinit_stroke3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                stroke_num,     /* stroke device number     */
    Pint                init_view_ind,  /* initial view index       */
    const Ppoint_list3  *init_stroke,   /* initial stroke           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Pstroke_data3 *stroke_data    /* data record              */
);
```

INITIALIZE STROKE

```
void pinit_stroke(
    Pint                ws_id,          /* workstation identifier   */
    Pint                stroke_num,     /* stroke device number     */
    Pint                init_view_ind,  /* initial view index       */
    const Ppoint_list   *init_stroke,   /* initial stroke           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Pstroke_data  *stroke_data    /* data record              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INITIALIZE VALUATOR 3

```
void pinit_val3(
    Pint              ws_id,        /* workstation identifier   */
    Pint              val_num,      /* valuator device number   */
    Pfloat            init_value,   /* initial value            */
    Pint              pet,          /* prompt and echo type     */
    const Plimit3     *echo_vol,    /* echo volume              */
    const Pval_data3  *val_data     /* data record              */
);
```

INITIALIZE VALUATOR

```
void pinit_val(
    Pint              ws_id,        /* workstation identifier   */
    Pint              val_num,      /* valuator device number   */
    Pfloat            init_value,   /* initial value            */
    Pint              pet,          /* prompt and echo type     */
    const Plimit      *echo_area,   /* echo area                */
    const Pval_data   *val_data     /* data record              */
);
```

INITIALIZE CHOICE 3

```
void pinit_choice3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                choice_num,     /* choice device number     */
    Pin_status          init_status,    /* initial choice status    */
    Pint                init_choice,    /* initial choice           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Pchoice_data3 *choice_data    /* data record              */
);
```

INITIALIZE CHOICE

```
void pinit_choice(
    Pint                ws_id,          /* workstation identifier   */
    Pint                choice_num,     /* choice device number     */
    Pin_status          init_status,    /* initial choice status    */
    Pint                init_choice,    /* initial choice           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Pchoice_data  *choice_data    /* data record              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INITIALIZE PICK 3

```
void pinit_pick3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                pick_num,       /* pick device number       */
    Pin_status          init_status,    /* initial pick status      */
    const Ppick_path    *init_pick,     /* initial pick path        */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Ppick_data3   *pick_data,     /* data record              */
    Ppath_order         order           /* pick path order          */
);
```

INITIALIZE PICK

```
void pinit_pick(
    Pint                ws_id,          /* workstation identifier   */
    Pint                pick_num,       /* pick device number       */
    Pin_status          init_status,    /* initial pick status      */
    const Ppick_path    *init_pick,     /* initial pick path        */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Ppick_data    *pick_data,     /* data record              */
    Ppath_order         order           /* pick path order          */
);
```

INITIALIZE STRING 3

```
void pinit_string3(
    Pint                  ws_id,        /* workstation identifier   */
    Pint                  string_num,   /* string device number     */
    const char            *init_string, /* initial string           */
    Pint                  pet,          /* prompt and echo type     */
    const Plimit3         *echo_vol,    /* echo volume              */
    const Pstring_data3   *string_data  /* data record              */
);
```

INITIALIZE STRING

```
void pinit_string(
    Pint                  ws_id,        /* workstation identifier   */
    Pint                  string_num,   /* string device number     */
    const char            *init_string, /* initial string           */
    Pint                  pet,          /* prompt and echo type     */
    const Plimit          *echo_area,   /* echo area                */
    const Pstring_data    *string_data  /* data record              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577449 064 ■

7.10.3 Setting the mode of input devices

SET LOCATOR MODE

```
void pset_loc_mode(
    Pint            ws_id,        /* workstation identifier   */
    Pint            loc_num,      /* locator device number    */
    Pop_mode        op_mode,      /* operating mode           */
    Pecho_switch    echo_switch   /* echo switch              */
);
```

SET STROKE MODE

```
void pset_stroke_mode(
    Pint            ws_id,        /* workstation identifier   */
    Pint            stroke_num,   /* stroke device number     */
    Pop_mode        op_mode,      /* operating mode           */
    Pecho_switch    echo_switch   /* echo switch              */
);
```

SET VALUATOR MODE

```
void pset_val_mode(
    Pint            ws_id,        /* workstation identifier   */
    Pint            val_num,      /* valuator device number   */
    Pop_mode        op_mode,      /* operating mode           */
    Pecho_switch    echo_switch   /* echo switch              */
);
```

SET CHOICE MODE

```
void pset_choice_mode(
    Pint            ws_id,        /* workstation identifier   */
    Pint            choice_num,   /* choice device number     */
    Pop_mode        op_mode,      /* operating mode           */
    Pecho_switch    echo_switch   /* echo switch              */
);
```

SET PICK MODE

```
void pset_pick_mode(
    Pint            ws_id,        /* workstation identifier   */
    Pint            pick_num,     /* pick device number       */
    Pop_mode        op_mode,      /* operating mode           */
    Pecho_switch    echo_switch   /* echo switch              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577450 886 ■

SET STRING MODE

```
void pset_string_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          string_num,   /* string device number     */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);
```

7.10.4 Request input functions

REQUEST LOCATOR 3

```
void preq_loc3(
    Pint          ws_id,        /* workstation identifier   */
    Pint          loc_num,      /* locator device number    */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint3       *loc_pos      /* OUT locator position     */
);
```

REQUEST LOCATOR

```
void preq_loc(
    Pint          ws_id,        /* workstation identifier   */
    Pint          loc_num,      /* locator device number    */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint        *loc_pos      /* OUT locator position     */
);
```

REQUEST STROKE 3

```
void preq_stroke3(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint_list3  *stroke       /* OUT stroke               */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE 3; the maximum size of a stroke point list supported by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA 3.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577451 712 ■

REQUEST STROKE

```
void preq_stroke(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint_list   *stroke       /* OUT stroke               */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE; the maximum size of a stroke point list supported by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA.

REQUEST VALUATOR

```
void preq_val(
    Pint          ws_id,        /* workstation identifier   */
    Pint          val_num,      /* valuator device number   */
    Pin_status    *in_status,   /* OUT input status         */
    Pfloat        *value        /* OUT valuator value       */
);
```

REQUEST CHOICE

```
void preq_choice(
    Pint          ws_id,        /* workstation identifier   */
    Pint          choice_num,   /* choice device number     */
    Pin_status    *in_status,   /* OUT choice status        */
    Pint          *choice       /* OUT requested choice     */
);
```

REQUEST PICK

```
void preq_pick(
    Pint          ws_id,        /* workstation identifier        */
    Pint          pick_num,     /* pick device number            */
    Pint          depth,        /* max. depth of returned path   */
    Pin_status    *in_status,   /* OUT pick status               */
    Ppick_path    *pick         /* OUT requested pick path       */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

REQUEST STRING

```
void preq_string(
    Pint        ws_id,        /* workstation identifier   */
    Pint        string_num,   /* string device number     */
    Pin_status  *in_status,   /* OUT input status         */
    char        *string       /* OUT requested string     */
);
```

The application shall allocate the memory for the character string returned by this function. The maximum size of the returned character string is specified by INITIALIZE STRING or INITIALIZE STRING 3; the maximum size of a character string supported by the implementation is returned by INQUIRE DEFAULT STRING DEVICE DATA or INQUIRE DEFAULT STRING DEVICE DATA 3.

7.10.5 Sample input functions

SAMPLE LOCATOR 3

```
void psample_loc3(
    Pint      ws_id,       /* workstation identifier   */
    Pint      loc_num,     /* locator device number    */
    Pint      *view_ind,   /* OUT view index           */
    Ppoint3   *loc_pos     /* OUT locator position     */
);
```

SAMPLE LOCATOR

```
void psample_loc(
    Pint     ws_id,       /* workstation identifier   */
    Pint     loc_num,     /* locator device number    */
    Pint     *view_ind,   /* OUT view index           */
    Ppoint   *loc_pos     /* OUT locator position     */
);
```

SAMPLE STROKE 3

```
void psample_stroke3(
    Pint           ws_id,        /* workstation identifier   */
    Pint           stroke_num,   /* stroke device number     */
    Pint           *view_ind,    /* OUT view index           */
    Ppoint_list3   *stroke       /* OUT stroke               */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE 3; the maximum size of a stroke point list supported by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA 3.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577453 595 ■

SAMPLE STROKE

```
void psample_stroke(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint_list   *stroke       /* OUT stroke               */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE; the maximum size of a stroke point list supported by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA.

SAMPLE VALUATOR

```
void psample_val(
    Pint     ws_id,     /* workstation identifier   */
    Pint     val_num,   /* valuator device number   */
    Pfloat   *value     /* OUT valuator value       */
);
```

SAMPLE CHOICE

```
void psample_choice(
    Pint         ws_id,              /* workstation identifier    */
    Pint         choice_num,         /* choice device number      */
    Pin_status   *choice_in_status,  /* OUT choice input status   */
    Pint         *choice             /* OUT choice                */
);
```

SAMPLE PICK

```
void psample_pick(
    Pint         ws_id,              /* workstation identifier        */
    Pint         pick_num,           /* pick device number            */
    Pint         depth,              /* max. depth of returned path   */
    Pin_status   *pick_in_status,    /* OUT pick input status         */
    Ppick_path   *pick               /* OUT pick path                 */
);
```

SAMPLE STRING

```
void psample_string(
    Pint    ws_id,        /* workstation identifier   */
    Pint    string_num,   /* string device number     */
    char    *string       /* OUT string               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The application shall allocate the memory for the character string returned by this function. The maximum size of the returned character string is specified by INITIALIZE STRING or INITIALIZE STRING 3; the maximum size of a character string supported by the implementation is returned by INQUIRE DEFAULT STRING DEVICE DATA or INQUIRE DEFAULT STRING DEVICE DATA 3.

7.10.6 Event input functions

AWAIT EVENT

```
void pawait_event(
    Pfloat      timeout,      /* timeout (seconds)                */
    Pint        *ws_id,       /* OUT workstation identifier       */
    Pin_class   *dev_class,   /* OUT device class                 */
    Pint        *in_num       /* OUT logical input device number  */
);
```

FLUSH DEVICE EVENTS

```
void pflush_events(
    Pint        ws_id,        /* workstation identifier       */
    Pin_class   dev_class,    /* device class                 */
    Pint        in_num        /* logical input device number  */
);
```

GET LOCATOR 3

```
void pget_loc3(
    Pint      *view_ind,   /* OUT view index         */
    Ppoint3   *loc_pos     /* OUT locator position   */
);
```

GET LOCATOR

```
void pget_loc(
    Pint     *view_ind,   /* OUT view index         */
    Ppoint   *loc_pos     /* OUT locator position   */
);
```

GET STROKE 3

```
void pget_stroke3(
    Pint           *view_ind,   /* OUT view index   */
    Ppoint_list3   *stroke      /* OUT stroke       */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE 3; the maximum size of a stroke point list supported

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577455 368 ■

by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA 3.

GET STROKE

```
void pget_stroke(
    Pint          *view_ind,   /* OUT view index   */
    Ppoint_list   *stroke      /* OUT stroke       */
);
```

The application shall allocate the memory for the point list returned by this function. The maximum size of the returned stroke point list is specified by INITIALIZE STROKE; the maximum size of a stroke point list supported by the implementation is returned by INQUIRE DEFAULT STROKE DEVICE DATA.

GET VALUATOR

```
void pget_val(
    Pfloat   *value   /* OUT valuator value   */
);
```

GET CHOICE

```
void pget_choice(
    Pin_status   *in_status,   /* OUT choice status   */
    Pint         *choice       /* OUT choice          */
);
```

GET PICK

```
void pget_pick(
    Pint         depth,        /* max. depth of returned path   */
    Pin_status   *in_status,   /* OUT pick status               */
    Ppick_path   *pick         /* OUT pick path                 */
);
```

GET STRING

```
void pget_string(
    char    *string   /* OUT string   */
);
```

The application shall allocate the memory for the character string returned by this function. The maximum size of the returned character string is specified by INITIALIZE STRING or INITIALIZE STRING 3; the maximum size of a character string supported by the implementation is returned by INQUIRE DEFAULT STRING DEVICE DATA or INQUIRE DEFAULT STRING DEVICE DATA 3.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.11 Metafile functions

WRITE ITEM TO METAFILE

```
void pwrite_item(
    Pint                ws_id,              /* workstation identifier    */
    Pint                item_type,          /* item type                 */
    Pint                item_data_length,   /* item data record length   */
    const Pitem_data    *item_data          /* item data record          */
);
```

GET ITEM TYPE FROM METAFILE

```
void pget_item_type(
    Pint   ws_id,               /* workstation identifier       */
    Pint   *item_type,          /* OUT item type                */
    Pint   *item_data_length    /* OUT item data record length  */
);
```

READ ITEM FROM METAFILE

```
void pread_item(
    Pint         ws_id,                 /* workstation identifier       */
    Pint         max_item_data_length,  /* max item data record length  */
    Pitem_data   *item_data             /* OUT item data record         */
);
```

INTERPRET ITEM

```
void pinterpret_item(
    Pint                type,               /* item type                 */
    Pint                item_data_length,   /* item data record length   */
    const Pitem_data    *item_data          /* item data record          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.12 Inquiry functions

7.12.1 Inquiry functions for operating state values

INQUIRE SYSTEM STATE VALUE

```
void pinq_sys_st(
    Psys_st   *sys_st   /* OUT the system state   */
);
```

INQUIRE WORKSTATION STATE VALUE

```
void pinq_ws_st(
    Pws_st   *ws_st   /* OUT workstation state   */
);
```

INQUIRE STRUCTURE STATE VALUE

```
void pinq_struct_st(
    Pstruct_st   *struct_st   /* OUT structure state   */
);
```

INQUIRE ARCHIVE STATE VALUE

```
void pinq_ar_st(
    Par_st   *ar_st   /* OUT archive state   */
);
```

7.12.2 Inquiry functions for PHIGS description table

INQUIRE LIST OF AVAILABLE WORKSTATION TYPES

```
void pinq_list_avail_ws_types(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *types,                /* OUT list of ws types       */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PHIGS FACILITIES

```
void pinq_phigs_facs(
    Pint        num_elems_appl_list,    /* # elems in appl list                        */
    Pint        start_ind,              /* starting index                              */
    Pint        *err_ind,               /* OUT error indicator                         */
    Pint        *max_open_ws,           /* OUT max. # simult. open ws                  */
    Pint        *max_open_ar,           /* OUT max. # simult. open archive files       */
    Pint        *num_avail_names,       /* OUT # available names for name sets         */
    Pint_list   *char_sets,             /* OUT list of character sets                  */
    Pint        *num_elems_impl_list,   /* OUT # elems in impl list                    */
    Pint        *iss_norm_max,          /* OUT ISS max. length of normal filter list   */
    Pint        *iss_inv_max            /* OUT ISS max. length of inverted filter list */
);
```

INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES

```
void pinq_gse_facs(
    Pint              num_elems_appl_list,   /* # elems in appl list                */
    Pint              start_ind,             /* starting index                      */
    Pint              *err_ind,              /* OUT error indicator                 */
    Pgse_id_dep_list  *gse,                  /* OUT list of GSE ids and dependencies */
    Pint              *num_elems_impl_list   /* OUT # elems in impl list            */
);
```

INQUIRE MODELLING CLIPPING FACILITIES

```
void pinq_model_clip_facs(
    Pint        num_elems_appl_list,   /* # elems in appl list          */
    Pint        start_ind,             /* starting index                */
    Pint        *err_ind,              /* OUT error indicator           */
    Pint        *num_planes,           /* OUT number of distinct planes */
    Pint_list   *ops,                  /* OUT list of operators         */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

7.12.3 Inquiry functions for PHIGS state list

INQUIRE EDIT MODE

```
void pinq_edit_mode(
    Pint         *err_ind,    /* OUT error indicator   */
    Pedit_mode   *edit_mode   /* OUT edit mode         */
);
```

INQUIRE SET OF OPEN WORKSTATIONS

```
void pinq_open_wss(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *open_ws_ids,          /* OUT list of ws ids         */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);
```

INQUIRE STRUCTURE IDENTIFIERS

```
void pinq_struct_ids(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *struct_ids,           /* OUT list of structure ids  */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);
```

INQUIRE ARCHIVE FILES

```
void pinq_ar_files(
    Pstore          store,       /* handle to Store object                  */
    Pint            *err_ind,    /* OUT error indicator                     */
    Par_file_list   **ar_files   /* OUT list of archive file names and ids  */
);
```

NOTE — The memory referenced by `*ar_files` is managed by `store`.

INQUIRE CONFLICT RESOLUTION

```
void pinq_conf_res(
    Pint        *err_ind,       /* OUT error indicator       */
    Pconf_res   *archive_res,   /* OUT archival resolution   */
    Pconf_res   *retrieve_res   /* OUT retrieval resolution  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE ALL CONFLICTING STRUCTURES

```
void pinq_all_conf_structs(
    Pint        ar_id,                  /* archive identifier                    */
    Pint        num_elems_appl_list,    /* # elems in appl list                  */
    Pint        start_ind,              /* starting index                        */
    Pint        *err_ind,               /* OUT error indicator                   */
    Pint_list   *ids,                   /* OUT list of conflicting structure ids */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list              */
);
```

INQUIRE CONFLICTING STRUCTURES IN NETWORK

```
void pinq_conf_structs_net(
    Pint                ar_id,                  /* archive identifier             */
    Pint                struct_id,              /* structure identifier           */
    Pstruct_net_source  source,                 /* structure network source       */
    Pint                num_elems_appl_list,    /* # elems in appl list           */
    Pint                start_ind,              /* starting index                 */
    Pint                *err_ind,               /* OUT error indicator            */
    Pint_list           *ids,                   /* OUT conflicting struct id list */
    Pint                *num_elems_impl_list    /* OUT # elems in impl list       */
);
```

INQUIRE MORE SIMULTANEOUS EVENTS

```
void pinq_more_simult_events(
    Pint                  *err_ind,        /* OUT error indicator       */
    Pmore_simult_events   *simult_events   /* OUT simultaneous events   */
);
```

7.12.4 Inquiry functions for workstation state list

INQUIRE WORKSTATION CONNECTION AND TYPE

```
void pinq_ws_conn_type(
    Pint    ws_id,      /* workstation identifier     */
    Pstore  store,      /* handle to Store object     */
    Pint    *err_ind,   /* OUT error indicator        */
    void    **conn_id,  /* OUT connection identifier  */
    Pint    *ws_type    /* OUT workstation type       */
);
```

NOTE — The memory referenced by `*conn_id` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577461 661 ■

INQUIRE LIST OF VIEW INDICES

```
void pinq_list_view_inds(
    Pint        ws_id,                  /* workstation identifier     */
    Pint        num_elems_appl_list,    /* # elems in appl list       */
    Pint        start_ind,              /* starting index             */
    Pint        *err_ind,               /* OUT error indicator        */
    Pint_list   *view_inds,             /* OUT list of view indices   */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list   */
);
```

INQUIRE VIEW REPRESENTATION

```
void pinq_view_rep(
    Pint         ws_id,       /* workstation identifier              */
    Pint         view_ind,    /* view index                          */
    Pint         *err_ind,    /* OUT error indicator                 */
    Pupd_st      *upd_st,     /* OUT transformation update state     */
    Pview_rep3   *cur_view,   /* OUT current view representation     */
    Pview_rep3   *req_view    /* OUT requested view representation   */
);
```

INQUIRE HLHSR MODE

```
void pinq_hlhsr_mode(
    Pint      ws_id,       /* workstation identifier    */
    Pint      *err_ind,    /* OUT error indicator       */
    Pupd_st   *upd_st,     /* OUT HLHSR update state    */
    Pint      *cur_mode,   /* OUT current HLHSR mode    */
    Pint      *req_mode    /* OUT requested HLHSR mode  */
);
```

INQUIRE POSTED STRUCTURES

```
void pinq_posted_structs(
    Pint                  ws_id,                /* workstation identifier          */
    Pint                  num_elems_appl_list,  /* # elems in appl list            */
    Pint                  start_ind,            /* starting index                  */
    Pint                  *err_ind,             /* OUT error indicator             */
    Pposted_struct_list   *struct_ids,          /* OUT list of posted structures   */
    Pint                  *num_elems_impl_list  /* OUT # elems in impl list        */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DISPLAY UPDATE STATE

```
void pinq_disp_upd_st(
    Pint              ws_id,              /* workstation identifier              */
    Pint              *err_ind,           /* OUT error indicator                 */
    Pdefer_mode       *def_mode,          /* OUT deferral mode                   */
    Pmod_mode         *mod_mode,          /* OUT modification mode               */
    Pdisp_surf_empty  *disp_surf_empty,   /* OUT display surface empty           */
    Pvisual_st        *vis_st             /* OUT state of visual representation  */
);
```

INQUIRE LIST OF POLYLINE INDICES

```
void pinq_list_line_inds(
    Pint        ws_id,                /* workstation identifier                */
    Pint        num_elems_appl_list,  /* # elems in appl list                  */
    Pint        start_ind,            /* starting index                        */
    Pint        *err_ind,             /* OUT error indicator                   */
    Pint_list   *def_line_ind,        /* OUT list of defined polyline indices  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list              */
);
```

INQUIRE POLYLINE REPRESENTATION

```
void pinq_line_rep(
    Pint          ws_id,      /* workstation identifier       */
    Pint          index,      /* polyline index               */
    Pinq_type     type,       /* type of returned value       */
    Pint          *err_ind,   /* OUT error indicator          */
    Pline_bundle  *line_rep   /* OUT polyline representation  */
);
```

INQUIRE LIST OF POLYMARKER INDICES

```
void pinq_list_marker_inds(
    Pint        ws_id,                /* workstation identifier                */
    Pint        num_elems_appl_list,  /* # elems in appl list                  */
    Pint        start_ind,            /* starting index                        */
    Pint        *err_ind,             /* OUT error indicator                   */
    Pint_list   *def_marker_ind,      /* OUT list of defined polymarker indices */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577463 434 ■

INQUIRE POLYMARKER REPRESENTATION

```
void pinq_marker_rep(
    Pint            ws_id,        /* workstation identifier          */
    Pint            index,        /* polymarker index                */
    Pinq_type       type,         /* type of returned value          */
    Pint            *err_ind,     /* OUT error indicator             */
    Pmarker_bundle  *marker_rep   /* OUT polymarker representation   */
);
```

INQUIRE LIST OF TEXT INDICES

```
void pinq_list_text_inds(
    Pint        ws_id,                /* workstation identifier            */
    Pint        num_elems_appl_list,  /* # elems in appl list              */
    Pint        start_ind,            /* starting index                    */
    Pint        *err_ind,             /* OUT error indicator               */
    Pint_list   *def_text_ind,        /* OUT list of defined text indices  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list          */
);
```

INQUIRE TEXT REPRESENTATION

```
void pinq_text_rep(
    Pint          ws_id,      /* workstation identifier   */
    Pint          index,      /* text index               */
    Pinq_type     type,       /* type of returned value   */
    Pint          *err_ind,   /* OUT error indicator      */
    Ptext_bundle  *text_rep   /* OUT text representation  */
);
```

INQUIRE LIST OF INTERIOR INDICES

```
void pinq_list_int_inds(
    Pint        ws_id,                /* workstation identifier                */
    Pint        num_elems_appl_list,  /* # elems in appl list                  */
    Pint        start_ind,            /* starting index                        */
    Pint        *err_ind,             /* OUT error indicator                   */
    Pint_list   *def_int_ind,         /* OUT list of defined interior indices  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE INTERIOR REPRESENTATION

```
void pinq_int_rep(
    Pint          ws_id,      /* workstation identifier       */
    Pint          index,      /* interior index               */
    Pinq_type     type,       /* type of returned value       */
    Pint          *err_ind,   /* OUT error indicator          */
    Pint_bundle   *int_rep    /* OUT interior representation  */
);
```

INQUIRE LIST OF EDGE INDICES

```
void pinq_list_edge_inds(
    Pint        ws_id,                /* workstation identifier           */
    Pint        num_elems_appl_list,  /* # elems in appl list             */
    Pint        start_ind,            /* starting index                   */
    Pint        *err_ind,             /* OUT error indicator              */
    Pint_list   *def_edge_ind,        /* OUT list of defined edge indices */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list         */
);
```

INQUIRE EDGE REPRESENTATION

```
void pinq_edge_rep(
    Pint          ws_id,      /* workstation identifier   */
    Pint          index,      /* edge index               */
    Pinq_type     type,       /* type of returned value   */
    Pint          *err_ind,   /* OUT error indicator      */
    Pedge_bundle  *edge_rep   /* OUT edge representation  */
);
```

INQUIRE LIST OF PATTERN INDICES

```
void pinq_list_pat_inds(
    Pint        ws_id,                /* workstation identifier           */
    Pint        num_elems_appl_list,  /* # elems in appl list             */
    Pint        start_ind,            /* starting index                   */
    Pint        *err_ind,             /* OUT error indicator              */
    Pint_list   *def_pat_ind,         /* OUT list of defined pattern indices */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list         */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577465 207 ■

INQUIRE PATTERN REPRESENTATION

```
void pinq_pat_rep(
    Pint        ws_id,      /* workstation identifier       */
    Pint        index,      /* pattern index                */
    Pinq_type   type,       /* type of returned value       */
    Pstore      store,      /* handle to Store object       */
    Pint        *err_ind,   /* OUT error indicator          */
    Ppat_rep    **pat_rep   /* OUT pattern representation   */
);
```

NOTE — The memory referenced by `*pat_rep` is managed by `store`.

INQUIRE COLOUR MODEL

```
void pinq_colr_model(
    Pint   ws_id,       /* workstation identifier     */
    Pint   *err_ind,    /* OUT error indicator        */
    Pint   *model       /* OUT current colour model   */
);
```

INQUIRE LIST OF COLOUR INDICES

```
void pinq_list_colr_inds(
    Pint        ws_id,                  /* workstation identifier       */
    Pint        num_elems_appl_list,    /* # elems in appl list         */
    Pint        start_ind,              /* starting index               */
    Pint        *err_ind,               /* OUT error indicator          */
    Pint_list   *colr_ind,              /* OUT list of colour indices   */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list     */
);
```

INQUIRE COLOUR REPRESENTATION

```
void pinq_colr_rep(
    Pint        ws_id,      /* workstation identifier     */
    Pint        colr_ind,   /* colour index               */
    Pinq_type   type,       /* type of returned value     */
    Pint        *err_ind,   /* OUT error indicator        */
    Pcolr_rep   *colr_rep   /* OUT colour representation  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE HIGHLIGHTING FILTER

```
void pinq_highl_filter(
    Pint      ws_id,          /* workstation identifier    */
    Pstore    store,          /* handle to Store object    */
    Pint      *err_ind,       /* OUT error indicator       */
    Pfilter   **highl_filter  /* OUT highlighting filter   */
);
```

NOTE — The memory referenced by `*highl_filter` is managed by `store`.

INQUIRE INVISIBILITY FILTER

```
void pinq_invis_filter(
    Pint      ws_id,          /* workstation identifier    */
    Pstore    store,          /* handle to Store object    */
    Pint      *err_ind,       /* OUT error indicator       */
    Pfilter   **invis_filter  /* OUT invisibility filter   */
);
```

NOTE — The memory referenced by `*invis_filter` is managed by `store`.

INQUIRE WORKSTATION TRANSFORMATION 3

```
void pinq_ws_tran3(
    Pint      ws_id,          /* workstation identifier              */
    Pint      *err_ind,       /* OUT error indicator                 */
    Pupd_st   *upd_st,        /* OUT update state                    */
    Plimit3   *req_win_lim,   /* OUT requested workstation window    */
    Plimit3   *cur_win_lim,   /* OUT current workstation window      */
    Plimit3   *req_vp_lim,    /* OUT requested workstation viewport  */
    Plimit3   *cur_vp_lim     /* OUT current workstation viewport    */
);
```

INQUIRE WORKSTATION TRANSFORMATION

```
void pinq_ws_tran(
    Pint      ws_id,          /* workstation identifier              */
    Pint      *err_ind,       /* OUT error indicator                 */
    Pupd_st   *upd_st,        /* OUT update state                    */
    Plimit    *req_win_lim,   /* OUT requested workstation window    */
    Plimit    *cur_win_lim,   /* OUT current workstation window      */
    Plimit    *req_vp_lim,    /* OUT requested workstation viewport  */
    Plimit    *cur_vp_lim     /* OUT current workstation viewport    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577467 08T ■

INQUIRE LOCATOR DEVICE STATE 3

```
void pinq_loc_st3(
    Pint           ws_id,          /* workstation identifier          */
    Pint           loc_num,        /* locator device number           */
    Pinq_type      type,           /* type of returned value          */
    Pstore         store,          /* handle to Store object          */
    Pint           *err_ind,       /* OUT error indicator             */
    Pop_mode       *op_mode,       /* OUT operating mode              */
    Pecho_switch   *echo_switch,   /* OUT echo switch                 */
    Pint           *init_view_ind, /* OUT initial view index          */
    Ppoint3        *init_loc_pos,  /* OUT initial locator position    */
    Pint           *prompt_echo,   /* OUT prompt/echo type            */
    Plimit3        *echo_vol,      /* OUT echo volume                 */
    Ploc_data3     **loc_data      /* OUT data record                 */
);
```

NOTE — The memory referenced by `*loc_data` is managed by `store`.

INQUIRE LOCATOR DEVICE STATE

```
void pinq_loc_st(
    Pint           ws_id,          /* workstation identifier          */
    Pint           loc_num,        /* locator device number           */
    Pinq_type      type,           /* type of returned value          */
    Pstore         store,          /* handle to Store object          */
    Pint           *err_ind,       /* OUT error indicator             */
    Pop_mode       *op_mode,       /* OUT operating mode              */
    Pecho_switch   *echo_switch,   /* OUT echo switch                 */
    Pint           *init_view_ind, /* OUT initial view index          */
    Ppoint         *init_loc_pos,  /* OUT initial locator position    */
    Pint           *prompt_echo,   /* OUT prompt/echo type            */
    Plimit         *echo_area,     /* OUT echo area                   */
    Ploc_data      **loc_data      /* OUT data record                 */
);
```

NOTE — The memory referenced by `*loc_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE STROKE DEVICE STATE 3

```
void pinq_stroke_st3(
    Pint            ws_id,          /* workstation identifier   */
    Pint            stroke_num,     /* stroke device number     */
    Pinq_type       type,           /* type of returned value   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pint            *init_view_ind, /* OUT initial view index   */
    Ppoint_list3    **init_stroke,  /* OUT initial stroke       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit3         *echo_vol,      /* OUT echo volume          */
    Pstroke_data3   **stroke_data   /* OUT data record          */
);
```

NOTE — The memory referenced by `*init_stroke` and `*stroke_data` is managed by `store`.

INQUIRE STROKE DEVICE STATE

```
void pinq_stroke_st(
    Pint            ws_id,          /* workstation identifier   */
    Pint            stroke_num,     /* stoke device number      */
    Pinq_type       type,           /* type of returned value   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pint            *init_view_ind, /* OUT initial view index   */
    Ppoint_list     **init_stroke,  /* OUT initial stroke       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit          *echo_area,     /* OUT echo area            */
    Pstroke_data    **stroke_data   /* OUT data record          */
);
```

NOTE — The memory referenced by `*init_stroke` and `*stroke_data` is managed by `store`.

INQUIRE VALUATOR DEVICE STATE 3

```
void pinq_val_st3(
    Pint            ws_id,          /* workstation identifier   */
    Pint            val_num,        /* valuator device number   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pfloat          *init_value,    /* OUT initial value        */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit3         *echo_vol,      /* OUT echo volume          */
    Pval_data3      **val_data      /* OUT data record          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577469 952 ■

NOTE — The memory referenced by `*val_data` is managed by `store`.

INQUIRE VALUATOR DEVICE STATE

```
void pinq_val_st(
    Pint          ws_id,          /* workstation identifier   */
    Pint          val_num,        /* valuator device number   */
    Pstore        store,          /* handle to Store object   */
    Pint          *err_ind,       /* OUT error indicator      */
    Pop_mode      *op_mode,       /* OUT operating mode       */
    Pecho_switch  *echo_switch,   /* OUT echo switch          */
    Pfloat        *init_value,    /* OUT initial value        */
    Pint          *prompt_echo,   /* OUT prompt/echo type     */
    Plimit        *echo_area,     /* OUT echo area            */
    Pval_data     **val_data      /* OUT data record          */
);
```

NOTE — The memory referenced by `*val_data` is managed by `store`.

INQUIRE CHOICE DEVICE STATE 3

```
void pinq_choice_st3(
    Pint          ws_id,          /* workstation identifier      */
    Pint          choice_num,     /* choice device number        */
    Pstore        store,          /* handle to Store object      */
    Pint          *err_ind,       /* OUT error indicator         */
    Pop_mode      *op_mode,       /* OUT operating mode          */
    Pecho_switch  *echo_switch,   /* OUT echo switch             */
    Pin_status    *init_status,   /* OUT initial choice status   */
    Pint          *init_choice,   /* OUT initial choice          */
    Pint          *prompt_echo,   /* OUT prompt/echo type        */
    Plimit3       *echo_vol,      /* OUT echo volume             */
    Pchoice_data3 **choice_data   /* OUT data record             */
);
```

NOTE — The memory referenced by `*choice_data` is managed by `store`.

INQUIRE CHOICE DEVICE STATE

```
void pinq_choice_st(
    Pint          ws_id,          /* workstation identifier      */
    Pint          choice_num,     /* choice device number        */
    Pstore        store,          /* handle to Store object      */
    Pint          *err_ind,       /* OUT error indicator         */
    Pop_mode      *op_mode,       /* OUT operating mode          */
    Pecho_switch  *echo_switch,   /* OUT echo switch             */
    Pin_status    *init_status,   /* OUT initial choice status   */
    Pint          *init_choice,   /* OUT initial choice          */
    Pint          *prompt_echo,   /* OUT prompt/echo type        */
    Plimit        *echo_area,     /* OUT echo area               */
    Pchoice_data  **choice_data   /* OUT data record             */
);
```

NOTE — The memory referenced by `*choice_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PICK DEVICE STATE 3

```
void pinq_pick_st3(
    Pint          ws_id,          /* workstation identifier    */
    Pint          pick_num,       /* pick device number        */
    Pinq_type     type,           /* type of returned value    */
    Pstore        store,          /* handle to Store object    */
    Pint          *err_ind,       /* OUT error indicator       */
    Pop_mode      *op_mode,       /* OUT operating mode        */
    Pecho_switch  *echo_switch,   /* OUT echo switch           */
    Pfilter       **pick_filter,  /* OUT pick filter           */
    Pin_status    *init_status,   /* OUT initial pick status   */
    Ppick_path    **init_pick,    /* OUT initial pick path     */
    Pint          *prompt_echo,   /* OUT prompt/echo type      */
    Plimit3       *echo_vol,      /* OUT echo volume           */
    Ppick_data3   **pick_data,    /* OUT data record           */
    Ppath_order   *order          /* OUT pick path order       */
);
```

NOTE — The memory referenced by `*pick_filter`, `*init_path` and `*pick_data` is managed by `store`.

INQUIRE PICK DEVICE STATE

```
void pinq_pick_st(
    Pint          ws_id,          /* workstation identifier    */
    Pint          pick_num,       /* pick device number        */
    Pinq_type     type,           /* type of returned value    */
    Pstore        store,          /* handle to Store object    */
    Pint          *err_ind,       /* OUT error indicator       */
    Pop_mode      *op_mode,       /* OUT operating mode        */
    Pecho_switch  *echo_switch,   /* OUT echo switch           */
    Pfilter       **pick_filter,  /* OUT pick filter           */
    Pin_status    *init_status,   /* OUT initial pick status   */
    Ppick_path    **init_pick,    /* OUT initial pick path     */
    Pint          *prompt_echo,   /* OUT prompt/echo type      */
    Plimit        *echo_area,     /* OUT echo area             */
    Ppick_data    **pick_data,    /* OUT data record           */
    Ppath_order   *order          /* OUT pick path order       */
);
```

NOTE — The memory referenced by `*pick_filter`, `*init_path` and `*pick_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577471 500 ■

INQUIRE STRING DEVICE STATE 3

```
void pinq_string_st3(
    Pint            ws_id,          /* workstation identifier   */
    Pint            string_num,     /* string device number     */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    char            **init_string,  /* OUT initial string       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit3         *echo_vol,      /* OUT echo volume          */
    Pstring_data3   **string_data   /* OUT data record          */
);
```

NOTE — The memory referenced by `*init_string` and `*string_data` is managed by `store`.

INQUIRE STRING DEVICE STATE

```
void pinq_string_st(
    Pint            ws_id,          /* workstation identifier   */
    Pint            string_num,     /* string device number     */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    char            **init_string,  /* OUT initial string       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit          *echo_area,     /* OUT echo area            */
    Pstring_data    **string_data   /* OUT data record          */
);
```

NOTE — The memory referenced by `*init_string` and `*string_data` is managed by `store`.

7.12.5 Inquiry functions for workstation description table

INQUIRE WORKSTATION CATEGORY

```
void pinq_ws_cat(
    Pint        ws_type,    /* workstation type             */
    Pint        *err_ind,   /* OUT error indicator          */
    Pws_cat     *cat        /* OUT workstation category     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DISPLAY SPACE SIZE 3

```
void pinq_disp_space_size3(
    Pint                ws_type,    /* workstation type      */
    Pint                *err_ind,   /* OUT error indicator   */
    Pdisp_space_size3   *size       /* OUT display size      */
);
```

INQUIRE DISPLAY SPACE SIZE

```
void pinq_disp_space_size(
    Pint                ws_type,    /* workstation type      */
    Pint                *err_ind,   /* OUT error indicator   */
    Pdisp_space_size    *size       /* OUT display size      */
);
```

The PHIGS abstract function INQUIRE HLHSR FACILITIES has been split into the following two functions.

INQUIRE HLHSR IDENTIFIER FACILITIES

```
void pinq_hlhsr_id_facs(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *hlhsr_ids,             /* OUT list of HLHSR ids     */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);
```

INQUIRE HLHSR MODE FACILITIES

```
void pinq_hlhsr_mode_facs(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *hlhsr_modes,           /* OUT list of HLHSR modes   */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);
```

INQUIRE VIEW FACILITIES

```
void pinq_view_facs(
    Pint   ws_type,         /* workstation type                      */
    Pint   *err_ind,        /* OUT error indicator                   */
    Pint   *num_view_ind    /* OUT number of predefined view indices */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577473 383 ■

INQUIRE PREDEFINED VIEW REPRESENTATION

```
void pinq_pred_view_rep(
    Pint         ws_type,    /* workstation type          */
    Pint         index,      /* predefined view index     */
    Pint         *err_ind,   /* OUT error indicator       */
    Pview_rep3   *view       /* OUT view representation   */
);
```

INQUIRE WORKSTATION CLASSIFICATION

```
void pinq_ws_class(
    Pint        ws_type,    /* workstation type        */
    Pint        *err_ind,   /* OUT error indicator     */
    Pws_class   *ws_class   /* OUT workstation class   */
);
```

INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES

```
void pinq_dyns_ws_attrs(
    Pint             ws_type,    /* workstation type          */
    Pint             *err_ind,   /* OUT error indicator       */
    Pdyns_ws_attrs   *attr       /* OUT attributes dynamics   */
);
```

INQUIRE DEFAULT DISPLAY UPDATE STATE

```
void pinq_def_disp_upd_st(
    Pint          ws_type,     /* workstation type        */
    Pint          *err_ind,    /* OUT error indicator     */
    Pdefer_mode   *def_mode,   /* OUT deferral mode       */
    Pmod_mode     *mod_mode    /* OUT modification mode   */
);
```

INQUIRE POLYLINE FACILITIES

```
void pinq_line_facs(
    Pint         ws_type,               /* workstation type          */
    Pint         num_elems_appl_list,   /* # elems in appl list      */
    Pint         start_ind,             /* starting index            */
    Pint         *err_ind,              /* OUT error indicator       */
    Pline_facs   *fac,                  /* OUT polyline facilities   */
    Pint         *num_elems_impl_list   /* OUT # elems in impl list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PREDEFINED POLYLINE REPRESENTATION

```
void pinq_pred_line_rep(
    Pint          ws_type,    /* workstation type              */
    Pint          index,      /* predefined index              */
    Pint          *err_ind,   /* OUT error indicator           */
    Pline_bundle  *bundle     /* OUT predefined polyline rep   */
);
```

INQUIRE POLYMARKER FACILITIES

```
void pinq_marker_facs(
    Pint          ws_type,              /* workstation type          */
    Pint          num_elems_appl_list,  /* # elems in appl list      */
    Pint          start_ind,            /* starting index            */
    Pint          *err_ind,             /* OUT error indicator       */
    Pmarker_facs  *fac,                 /* OUT polymarker facilities */
    Pint          *num_elems_impl_list  /* OUT # elems in impl list  */
);
```

INQUIRE PREDEFINED POLYMARKER REPRESENTATION

```
void pinq_pred_marker_rep(
    Pint            ws_type,    /* workstation type              */
    Pint            index,      /* predefined index              */
    Pint            *err_ind,   /* OUT error indicator           */
    Pmarker_bundle  *bundle     /* OUT predefined polymarker rep */
);
```

INQUIRE TEXT FACILITIES

```
void pinq_text_facs(
    Pint         ws_type,              /* workstation type          */
    Pint         num_elems_appl_list,  /* # elems in appl list      */
    Pint         start_ind,            /* starting index            */
    Pint         *err_ind,             /* OUT error indicator       */
    Ptext_facs   *fac,                 /* OUT text facilities       */
    Pint         *num_elems_impl_list  /* OUT # elems in impl list  */
);
```

INQUIRE PREDEFINED TEXT REPRESENTATION

```
void pinq_pred_text_rep(
    Pint          ws_type,    /* workstation type          */
    Pint          index,      /* predefined index          */
    Pint          *err_ind,   /* OUT error indicator       */
    Ptext_bundle  *bundle     /* OUT predefined text rep   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577475 156 ■

INQUIRE ANNOTATION FACILITIES

```
void pinq_anno_facs(
    Pint        ws_type,                /* workstation type                          */
    Pint        num_elems_appl_list,    /* # elems in appl list                      */
    Pint        start_ind,              /* starting index                            */
    Pint        *err_ind,               /* OUT error indicator                       */
    Pint_list   *styles,                /* OUT list annotation styles                */
    Pint        *num_elems_impl_list,   /* OUT # elems in impl list                  */
    Pint        *num_anno_char_hts,     /* OUT number of anno. text char. heights    */
    Pfloat      *min_anno_char_ht,      /* OUT minimum anno. text char. height       */
    Pfloat      *max_anno_char_ht       /* OUT maximum anno. text char. height       */
);
```

INQUIRE TEXT EXTENT

```
void pinq_text_extent(
    Pint              ws_type,          /* workstation type          */
    Pint              text_font,        /* text font                 */
    Pfloat            char_expan,       /* char expansion factor     */
    Pfloat            char_space,       /* char spacing              */
    Pfloat            char_ht,          /* char height               */
    Ptext_path        text_path,        /* text path                 */
    Phor_text_align   hor_text_align,   /* horizontal text alignment */
    Pvert_text_align  ver_text_align,   /* vertical text alignment   */
    const char        *char_string,     /* character string          */
    Pint              *err_ind,         /* OUT error indicator       */
    Prect             *rect,            /* OUT extent rectangle      */
    Ppoint            *offset           /* OUT concatenation offset  */
);
```

INQUIRE INTERIOR FACILITIES

```
void pinq_int_facs(
    Pint        ws_type,                        /* workstation type                    */
    Pint        hatch_num_elems_appl_length,    /* # elems in appl hatch style list    */
    Pint        hatch_start_ind,                /* starting index                      */
    Pint        *err_ind,                       /* OUT error indicator                 */
    Pint_facs   *int_facs,                      /* OUT interior facilities             */
    Pint        *hatch_num_elems_impl_list      /* OUT # elems in impl list            */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE PREDEFINED INTERIOR REPRESENTATION

```
void pinq_pred_int_rep(
    Pint          ws_type,    /* workstation type             */
    Pint          index,      /* predefined index             */
    Pint          *err_ind,   /* OUT error indicator          */
    Pint_bundle   *bundle     /* OUT predefined interior rep  */
);
```

INQUIRE EDGE FACILITIES

```
void pinq_edge_facs(
    Pint         ws_type,               /* workstation type          */
    Pint         num_elems_appl_list,   /* # elems in appl list      */
    Pint         start_ind,             /* starting index            */
    Pint         *err_ind,              /* OUT error indicator       */
    Pedge_facs   *fac,                  /* OUT edge facilities       */
    Pint         *num_elems_impl_list   /* OUT # elems in impl list  */
);
```

INQUIRE PREDEFINED EDGE REPRESENTATION

```
void pinq_pred_edge_rep(
    Pint           ws_type,    /* workstation type          */
    Pint           index,      /* predefined index          */
    Pint           *err_ind,   /* OUT error indicator       */
    Pedge_bundle   *bundle     /* OUT predefined edge rep   */
);
```

INQUIRE PATTERN FACILITIES

```
void pinq_pat_facs(
    Pint   ws_type,    /* workstation type                          */
    Pint   *err_ind,   /* OUT error indicator                       */
    Pint   *num_pred   /* OUT number of predefined pattern indices  */
);
```

INQUIRE PREDEFINED PATTERN REPRESENTATION

```
void pinq_pred_pat_rep(
    Pint       ws_type,    /* workstation type           */
    Pint       index,      /* predefined index           */
    Pstore     store,      /* handle to Store object     */
    Pint       *err_ind,   /* OUT error indicator        */
    Ppat_rep   **pat_rep   /* OUT predefined pattern rep */
);
```

NOTE — The memory referenced by `*pat_rep` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE COLOUR MODEL FACILITIES

```
void pinq_colr_model_facs(
    Pint        ws_type,                /* workstation type           */
    Pint        num_elems_appl_list,    /* # elems in appl list       */
    Pint        start_ind,              /* starting index             */
    Pint        *err_ind,               /* OUT error indicator        */
    Pint_list   *models,                /* OUT list of colour models  */
    Pint        *num_elems_impl_list,   /* OUT # elems in impl list   */
    Pint        *def                    /* OUT default colour model   */
);
```

INQUIRE COLOUR FACILITIES

```
void pinq_colr_facs(
    Pint         ws_type,    /* workstation type       */
    Pint         *err_ind,   /* OUT error indicator    */
    Pcolr_facs   *fac        /* OUT colour facilities  */
);
```

INQUIRE PREDEFINED COLOUR REPRESENTATION

```
void pinq_pred_colr_rep(
    Pint        ws_type,    /* workstation type                        */
    Pint        colr_ind,   /* predefined index                        */
    Pint        *err_ind,   /* OUT error indicator                     */
    Pcolr_rep   *colr_rep   /* OUT predefined colour representation    */
);
```

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3

```
void pinq_list_avail_gdp3(
    Pint        ws_type,                /* workstation type           */
    Pint        num_elems_appl_list,    /* # elems in appl list       */
    Pint        start_ind,              /* starting index             */
    Pint        *err_ind,               /* OUT error indicator        */
    Pint_list   *gdps,                  /* OUT list of GDPs           */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES

```
void pinq_list_avail_gdp(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *gdps,                  /* OUT list of GDPs          */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);
```

INQUIRE GENERALIZED DRAWING PRIMITIVE 3

```
void pinq_gdp3(
    Pint     ws_type,     /* workstation type                */
    Pint     gdp,         /* GDP function identifier         */
    Pint     *err_ind,    /* OUT error indicator             */
    Pint     *num_attr,   /* OUT number of attributes used   */
    Pattrs   attr[5]      /* OUT list of attributes used     */
);
```

INQUIRE GENERALIZED DRAWING PRIMITIVE

```
void pinq_gdp(
    Pint     ws_type,     /* workstation type                */
    Pint     gdp,         /* GDP function identifier         */
    Pint     *err_ind,    /* OUT error indicator             */
    Pint     *num_attr,   /* OUT number of attributes used   */
    Pattrs   attr[5]      /* OUT list of attributes used     */
);
```

INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS

```
void pinq_list_avail_gse(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *gses,                  /* OUT list of GSEs          */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED

```
void pinq_num_disp_pris(
    Pint   ws_type,    /* workstation type                    */
    Pint   *err_ind,   /* OUT error indicator                 */
    Pint   *num_pri    /* OUT number of structure priorities  */
);
```

INQUIRE WORKSTATION STATE TABLE LENGTHS

```
void pinq_ws_st_table(
    Pint           ws_type,    /* workstation type                  */
    Pint           *err_ind,   /* OUT error indicator               */
    Pws_st_tables  *lengths    /* OUT lengths of workstation tables */
);
```

INQUIRE DYNAMICS OF STRUCTURES

```
void pinq_dyns_structs(
    Pint           ws_type,    /* workstation type        */
    Pint           *err_ind,   /* OUT error indicator     */
    Pdyns_structs  *dyns       /* OUT structure dynamics  */
);
```

INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES

```
void pinq_num_avail_in(
    Pint      ws_type,    /* workstation type             */
    Pint      *err_ind,   /* OUT error indicator          */
    Pnum_in   *num_in     /* OUT number of input devices  */
);
```

INQUIRE DEFAULT LOCATOR DEVICE DATA 3

```
void pinq_def_loc_data3(
    Pint         ws_type,      /* workstation type                */
    Pint         loc_num,      /* logical input device number     */
    Pstore       store,        /* handle to Store object          */
    Pint         *err_ind,     /* OUT error indicator             */
    Ppoint3      *loc_pos,     /* OUT default initial position    */
    Pint_list    **pet_list,   /* OUT list of prompt and echo types */
    Plimit3      *echo_vol,    /* OUT default echo volume         */
    Ploc_data3   **loc_data    /* OUT default data record         */
);
```

NOTE — The memory referenced by `*pet_list` and `*loc_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DEFAULT LOCATOR DEVICE DATA

```
void pinq_def_loc_data(
    Pint        ws_type,      /* workstation type                    */
    Pint        loc_num,      /* logical input device number         */
    Pstore      store,        /* handle to Store object              */
    Pint        *err_ind,     /* OUT error indicator                 */
    Ppoint      *loc_pos,     /* OUT default locator position        */
    Pint_list   **pet_list,   /* OUT list of prompt and echo types   */
    Plimit      *echo_area,   /* OUT default echo area               */
    Ploc_data   **loc_data    /* OUT default data record             */
);
```

NOTE — The memory referenced by `*pet_list` and `*loc_data` is managed by `store`.

INQUIRE DEFAULT STROKE DEVICE DATA 3

```
void pinq_def_stroke_data3(
    Pint            ws_type,        /* workstation type                  */
    Pint            stroke_num,     /* logical input device number       */
    Pstore          store,          /* handle to Store object            */
    Pint            *err_ind,       /* OUT error indicator               */
    Pint            *max_buf_size,  /* OUT max. input buffer size        */
    Pint_list       **pet_list,     /* OUT list of prompt and echo types */
    Plimit3         *echo_vol,      /* OUT default echo volume           */
    Pstroke_data3   **stroke_data   /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*stroke_data` is managed by `store`.

INQUIRE DEFAULT STROKE DEVICE DATA

```
void pinq_def_stroke_data(
    Pint          ws_type,        /* workstation type                  */
    Pint          stroke_num,     /* logical input device number       */
    Pstore        store,          /* handle to Store object            */
    Pint          *err_ind,       /* OUT error indicator               */
    Pint          *max_buf_size,  /* OUT max. input buffer size        */
    Pint_list     **pet_list,     /* OUT list of prompt and echo types */
    Plimit        *echo_area,     /* OUT default echo area             */
    Pstroke_data  **stroke_data   /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*stroke_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577481 45T ■

INQUIRE DEFAULT VALUATOR DEVICE DATA 3

```
void pinq_def_val_data3(
    Pint         ws_type,      /* workstation type                  */
    Pint         val_num,      /* logical input device number       */
    Pstore       store,        /* handle to Store object            */
    Pint         *err_ind,     /* OUT error indicator               */
    Pfloat       *def_value,   /* OUT default initial value         */
    Pint_list    **pet_list,   /* OUT list of prompt and echo types */
    Plimit3      *echo_vol,    /* OUT default echo volume           */
    Pval_data3   **val_data    /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*val_data` is managed by `store`.

INQUIRE DEFAULT VALUATOR DEVICE DATA

```
void pinq_def_val_data(
    Pint         ws_type,      /* workstation type                  */
    Pint         val_num,      /* logical input device number       */
    Pstore       store,        /* handle to Store object            */
    Pint         *err_ind,     /* OUT error indicator               */
    Pfloat       *def_value,   /* OUT default initial value         */
    Pint_list    **pet_list,   /* OUT list of prompt and echo types */
    Plimit       *echo_area,   /* OUT default echo area             */
    Pval_data    **val_data    /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*val_data` is managed by `store`.

INQUIRE DEFAULT CHOICE DEVICE DATA 3

```
void pinq_def_choice_data3(
    Pint             ws_type,       /* workstation type                  */
    Pint             choice_num,    /* logical input device number       */
    Pstore           store,         /* handle to Store object            */
    Pint             *err_ind,      /* OUT error indicator               */
    Pint             *max_choices,  /* OUT max. number of choices        */
    Pint_list        **pet_list,    /* OUT list of prompt and echo types */
    Plimit3          *echo_vol,     /* OUT default echo volume           */
    Pchoice_data3    **choice_data  /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*choice_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INQUIRE DEFAULT CHOICE DEVICE DATA

```
void pinq_def_choice_data(
    Pint          ws_type,       /* workstation type                  */
    Pint          choice_num,    /* logical input device number       */
    Pstore        store,         /* handle to Store object            */
    Pint          *err_ind,      /* OUT error indicator               */
    Pint          *max_choices,  /* OUT max. number of choices        */
    Pint_list     **pet_list,    /* OUT list of prompt and echo types */
    Plimit        *echo_area,    /* OUT default echo area             */
    Pchoice_data  **choice_data  /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*choice_data` is managed by `store`.

INQUIRE DEFAULT PICK DEVICE DATA 3

```
void pinq_def_pick_data3(
    Pint          ws_type,       /* workstation type                  */
    Pint          pick_num,      /* logical input device number       */
    Pstore        store,         /* handle to Store object            */
    Pint          *err_ind,      /* OUT error indicator               */
    Pint_list     **pet_list,    /* OUT list of prompt and echo types */
    Plimit3       *echo_vol,     /* OUT default echo volume           */
    Ppick_data3   **pick_data    /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*pick_data` is managed by `store`.

INQUIRE DEFAULT PICK DEVICE DATA

```
void pinq_def_pick_data(
    Pint          ws_type,       /* workstation type                  */
    Pint          pick_num,      /* logical input device number       */
    Pstore        store,         /* handle to Store object            */
    Pint          *err_ind,      /* OUT error indicator               */
    Pint_list     **pet_list,    /* OUT list of prompt and echo types */
    Plimit        *echo_area,    /* OUT default echo area             */
    Ppick_data    **pick_data    /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*pick_data` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

INQUIRE DEFAULT STRING DEVICE DATA 3

```
void pinq_def_string_data3(
    Pint            ws_type,        /* workstation type                  */
    Pint            string_num,     /* logical input device number       */
    Pstore          store,          /* handle to Store object            */
    Pint            *err_ind,       /* OUT error indicator               */
    Pint            *max_buf_size,  /* OUT max. input buffer size        */
    Pint_list       **pet_list,     /* OUT list of prompt and echo types */
    Plimit3         *echo_vol,      /* OUT default echo volume           */
    Pstring_data3   **string_data   /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*string_data` is managed by `store`.

INQUIRE DEFAULT STRING DEVICE DATA

```
void pinq_def_string_data(
    Pint            ws_type,        /* workstation type                  */
    Pint            string_num,     /* logical input device number       */
    Pstore          store,          /* handle to Store object            */
    Pint            *err_ind,       /* OUT error indicator               */
    Pint            *max_buf_size,  /* OUT max. input buffer size        */
    Pint_list       **pet_list,     /* OUT list of prompt and echo types */
    Plimit          *echo_area,     /* OUT default echo area             */
    Pstring_data    **string_data   /* OUT default data record           */
);
```

NOTE — The memory referenced by `*pet_list` and `*string_data` is managed by `store`.

7.12.6 Inquiry function for structure state list

INQUIRE SET OF WORKSTATIONS TO WHICH POSTED

```
void pinq_wss_posted(
    Pint        struct_id,              /* structure identifier      */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *ws,                    /* OUT list of workstations  */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.12.7 Inquiry functions for structure content

INQUIRE OPEN STRUCTURE

```
void pinq_open_struct(
    Pint                  *err_ind,    /* OUT error indicator           */
    Popen_struct_status   *status,     /* OUT status of open structure  */
    Pint                  *struct_id   /* OUT structure identifier      */
);
```

INQUIRE ELEMENT POINTER

```
void pinq_elem_ptr(
    Pint   *err_ind,         /* OUT error indicator         */
    Pint   *elem_ptr_value   /* OUT element pointer value   */
);
```

INQUIRE CURRENT ELEMENT TYPE AND SIZE

```
void pinq_cur_elem_type_size(
    Pint         *err_ind,     /* OUT error indicator   */
    Pelem_type   *elem_type,   /* OUT element type      */
    size_t       *elem_size    /* OUT element size      */
);
```

INQUIRE CURRENT ELEMENT CONTENT

```
void pinq_cur_elem_content(
    Pstore       store,        /* handle to Store object   */
    Pint         *err_ind,     /* OUT error indicator      */
    Pelem_data   **elem_data   /* OUT element data         */
);
```

NOTE — The memory referenced by `*elem_data` is managed by `store`.

INQUIRE ELEMENT TYPE AND SIZE

```
void pinq_elem_type_size(
    Pint         struct_id,    /* structure identifier   */
    Pint         elem_num,     /* element number         */
    Pint         *err_ind,     /* OUT error indicator    */
    Pelem_type   *elem_type,   /* OUT element type       */
    size_t       *elem_size    /* OUT element size       */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577485 0T5 ■

INQUIRE ELEMENT CONTENT

```
void pinq_elem_content(
    Pint          struct_id,    /* structure identifier     */
    Pint          elem_num,     /* element number           */
    Pstore        store,        /* handle to Store object   */
    Pint          *err_ind,     /* OUT error indicator      */
    Pelem_data    **elem_data   /* OUT data record          */
);
```

NOTE — The memory referenced by `*elem_data` is managed by `store`.

INQUIRE STRUCTURE STATUS

```
void pinq_struct_status(
    Pint             struct_id,   /* structure identifier   */
    Pint             *err_ind,    /* OUT error indicator    */
    Pstruct_status   *status      /* OUT existence status   */
);
```

INQUIRE PATHS TO ANCESTORS

```
void pinq_paths_ances(
    Pint                  struct_id,   /* structure identifier      */
    Ppath_order           order,       /* path order                */
    Pint                  depth,       /* path depth                */
    Pstore                store,       /* handle to Store object    */
    Pint                  *err_ind,    /* OUT error indicator       */
    Pelem_ref_list_list   **paths      /* OUT structure path list   */
);
```

NOTE — The memory referenced by `*paths` is managed by `store`.

INQUIRE PATHS TO DESCENDANTS

```
void pinq_paths_descs(
    Pint                  struct_id,   /* structure identifier      */
    Ppath_order           order,       /* path order                */
    Pint                  depth,       /* path depth                */
    Pstore                store,       /* handle to Store object    */
    Pint                  *err_ind,    /* OUT error indicator       */
    Pelem_ref_list_list   **paths      /* OUT structure path list   */
);
```

NOTE — The memory referenced by `*paths` is managed by `store`.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ELEMENT SEARCH

```
void pelem_search(
    Pint                    struct_id,        /* structure identifier        */
    Pint                    start_elem,       /* starting element position   */
    Psearch_dir             dir,              /* search direction            */
    const Pelem_type_list   *incl,            /* element incl. list          */
    const Pelem_type_list   *excl,            /* element excl. list          */
    Pint                    *err_ind,         /* OUT error indicator         */
    Psearch_status          *status,          /* OUT search status           */
    Pint                    *found_elem_ptr   /* OUT found element pointer   */
);
```

INCREMENTAL SPATIAL SEARCH 3

```
void pincr_spa_search3(
    const Ppoint3           *ref_pt,          /* search reference point      */
    Pfloat                  dist,             /* search distance             */
    const Pelem_ref_list    *sp,              /* starting path list          */
    Pclip_ind               clip_ind,         /* modeling clip indicator     */
    Pint                    ceiling,          /* search ceiling index        */
    const Pfilter_list      *norm,            /* normal filter list          */
    const Pfilter_list      *inv,             /* inverted filter list        */
    Pint                    num_elems_appl_list, /* # elems in appl list     */
    Pint                    start_ind,        /* starting index              */
    Pint                    *err_ind,         /* OUT error indicator         */
    Pelem_ref_list          *fp,              /* OUT found path              */
    Pint                    *num_elems_impl_list /* OUT # elems in impl list */
);
```

INCREMENTAL SPATIAL SEARCH

```
void pincr_spa_search(
    const Ppoint            *ref_pt,          /* search reference point      */
    Pfloat                  dist,             /* search distance             */
    const Pelem_ref_list    *sp,              /* starting path list          */
    Pclip_ind               clip_ind,         /* modeling clip indicator     */
    Pint                    ceiling,          /* search ceiling index        */
    const Pfilter_list      *norm,            /* normal filter list          */
    const Pfilter_list      *inv,             /* inverted filter list        */
    Pint                    num_elems_appl_list, /* # elems in appl list     */
    Pint                    start_ind,        /* starting index              */
    Pint                    *err_ind,         /* OUT error indicator         */
    Pelem_ref_list          *fp,              /* OUT found path              */
    Pint                    *num_elems_impl_list /* OUT # elems in impl list */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577487 978 ■

7.12.8 Inquiry functions for error state list

INQUIRE INPUT QUEUE OVERFLOW

```
void pinq_in_overf(
    Pint        *err_ind,    /* OUT error indicator          */
    Pint        *ws_id,      /* OUT workstation identifier   */
    Pin_class   *in_class,   /* OUT input class              */
    Pint        *in_num      /* OUT input device number      */
);
```

INQUIRE ERROR HANDLING MODE

```
void pinq_err_hand_mode(
    Pint        *err_ind,   /* OUT error indicator   */
    Perr_mode   *err_mode   /* OUT error mode        */
);
```

7.13 Error control

EMERGENCY CLOSE PHIGS

```
void pemergency_close_phigs(
        void
);
```

ERROR HANDLING

```
void perr_hand(
    Pint          error_num,    /* error number                                    */
    Pint          func_num,     /* identifier of function that detected the error  */
    const char    *error_file   /* name of error file                              */
);
```

ERROR LOGGING

```
void perr_log(
    Pint          error_num,    /* error number                                    */
    Pint          func_num,     /* identifier of function that detected the error  */
    const char    *error_file   /* name of error file                              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

SET ERROR HANDLING MODE

```
void pset_err_hand_mode(
    Perr_mode   error_mode   /* error handling mode   */
);
```

7.14 Special interfaces

ESCAPE

```
void pescape(
    Pint                    func_id,     /* escape function identifier        */
    const Pescape_in_data   *in_data,    /* input data for the function       */
    Pstore                  store,       /* handle to Store object            */
    Pescape_out_data        **out_data   /* OUT output data of the function   */
);
```

If an error is detected by this function, `*out_data` shall be set to NULL.

NOTE — The memory referenced by `*out_data` is managed by `store`.

7.15 Binding defined utility functions

SET ERROR HANDLING

```
void pset_err_hand(
    void   (*new_err_hand)(),   /* application's error handling function   */
    void   (**old_err_hand)()   /* OUT old error handling function         */
);
```

CREATE STORE

```
void pcreate_store(
    Pint     *err_ind,   /* OUT error indicator   */
    Pstore   *store      /* OUT new Store         */
);
```

DELETE STORE

```
void pdel_store(
    Pint     *err_ind,   /* OUT error indicator          */
    Pstore   *store      /* IN/OUT Store to be deleted   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577489 740 ■

Annex A

(informative)

Data types in compilation order and external functions

Annex A lists the data types of the binding in an order suitable for compilation and also lists an external declaration for each function.

```
#include <stddef.h>
/*
 * Note: the field "int impl_dep;" is a place-holder for
 * a implementation dependent field.
 */
```

A.1 Macro definitions

```
/*                              <0          Implementation Dependent Errors */

#define  PE_NO_ERROR            (0)      /* No Error */

/* State Errors */
#define  PE_NOT_PHCL            (1)      /* Ignoring function, function requires
                                            state (PHCL,WSCL,STCL,ARCL) */
#define  PE_NOT_PHOP            (2)      /* Ignoring function, function requires
                                            state (PHOP,*,*,*) */
#define  PE_NOT_WSOP            (3)      /* Ignoring function, function requires
                                            state (PHOP,WSOP,*,*) */
#define  PE_NOT_CL              (4)      /* Ignoring function, function requires
                                            state (PHOP,WSCL,STCL,ARCL) */
#define  PE_NOT_STOP            (5)      /* Ignoring function, function requires
                                            state (PHOP,*,STOP,*) */
#define  PE_NOT_STCL            (6)      /* Ignoring function, function requires
                                            state (PHOP,*,STCL,*) */
#define  PE_NOT_AROP            (7)      /* Ignoring function, function requires
                                            state (PHOP,*,*,AROP) */

/* Workstation Errors */
#define  PE_BAD_CONN_ID         (50)     /* Ignoring function, connection iden-
                                            tifier not recognized by the imple-
                                            mentation */
#define  PE_WS_TYPE             (51)     /* Ignoring function, this information
                                            is not yet available for this works-
                                            tation type; open a workstation of
                                            this type and use the specific works-
                                            tation type */
#define  PE_BAD_WS_TYPE         (52)     /* Ignoring function, workstation type
                                            not recognized by the implementation
                                            */
#define  PE_DUP_WS_ID           (53)     /* Ignoring function, workstation iden-
                                            tifier already is in use */
#define  PE_WS_NOT_OPEN         (54)     /* Ignoring function, the specified
                                            workstation is not open */
#define  PE_NO_OPEN_WS          (55)     /* Ignoring function, workstation cannot
                                            be opened for an implementation
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577490 462 ■


```
                                        dependent reason */
#define PE_WS_NOT_MO            (56)    /* Ignoring function, specified worksta-
                                           tion is not of category MO */
#define PE_WS_MI                (57)    /* Ignoring function, specified worksta-
                                           tion is of category MI */
#define PE_WS_NOT_MI            (58)    /* Ignoring function, specified worksta-
                                           tion is not of category MI */
#define PE_WS_NO_OUTPUT         (59)    /* Ignoring function, the specified
                                           workstation does not have output
                                           capability (i.e., the workstation
                                           category is neither OUTPUT, OUTIN,
                                           nor MO) */
#define PE_WS_NOT_OUTIN         (60)    /* Ignoring function, specified worksta-
                                           tion is not of category OUTIN */
#define PE_WS_NO_INPUT          (61)    /* Ignoring function, specified worksta-
                                           tion is neither of category INPUT nor
                                           of category OUTIN */
#define PE_WS_NOT_OUT           (62)    /* Ignoring function, specified worksta-
                                           tion is neither of category OUTPUT
                                           nor of category OUTIN */
#define PE_MAX_WS               (63)    /* Ignoring function, opening this
                                           workstation would exceed the maximum
                                           number of simultaneously open works-
                                           tations */
#define PE_NO_GDP               (64)    /* Ignoring function, the specified
                                           workstation type is not able to gen-
                                           erate the specified generalized draw-
                                           ing primitive */

/* Output Attribute Errors */
#define PE_BUN_IND_LT_1         (100)   /* Ignoring function, the bundle index
                                           value is less than one */
#define PE_REP_UNDEF            (101)   /* Ignoring function, the specified
                                           representation has not been defined
                                           */
#define PE_REP_NOT_PREDEF       (102)   /* Ignoring function, the specified
                                           representation has not been prede-
                                           fined on this workstation */
#define PE_MAX_BUN              (103)   /* Ignoring function, setting this bun-
                                           dle table entry would exceed the max-
                                           imum number of entries allowed in the
                                           workstation bundle table */
#define PE_BAD_LINETYPE         (104)   /* Ignoring function, the specified
                                           linetype is not available on the
                                           specified workstation */
#define PE_BAD_MARKER_TYPE      (105)   /* Ignoring function, the specified
                                           marker type is not available on the
                                           specified workstation */
#define PE_BAD_FONT             (106)   /* Ignoring function, the specified font
                                           is not available for the requested
                                           text precision on the specified
                                           workstation */
#define PE_BAD_EDGETYPE         (107)   /* Ignoring function, the specified
                                           edgetype is not available on the
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
                                        specified workstation */
#define PE_BAD_INT_STYLE      (108)   /* Ignoring function, the specified
                                         interior style is not available on
                                         the workstation */
#define PE_NO_PAT             (109)   /* Ignoring function, interior style
                                         PATTERN is not supported on the
                                         workstation */
#define PE_BAD_COLR_MODEL     (110)   /* Ignoring function, the specified
                                         colour model is not available on the
                                         workstation */
#define PE_BAD_HLHSR_MODE     (111)   /* Ignoring function, the specified
                                         HLHSR mode is not available on the
                                         specified workstation */
#define PE_PAT_IND_LT_1       (112)   /* Ignoring function, the pattern index
                                         value is less than one */
#define PE_COLR_IND_LT_0      (113)   /* Ignoring function, the colour index
                                         value is less than zero */
#define PE_VIEW_IND_LT_0      (114)   /* Ignoring function, the view index
                                         value is less than zero */
#define PE_VIEW_IND_LT_1      (115)   /* Ignoring function, the view index
                                         value is less than one */
#define PE_BAD_PAT_DIM        (116)   /* Ignoring function, one of the dimen-
                                         sions of pattern colour array is less
                                         than one */
#define PE_BAD_COLR_DIM       (117)   /* Ignoring function, one of the dimen-
                                         sions of the colour index array is
                                         less than zero */
#define PE_BAD_COLR           (118)   /* Ignoring function, one of the com-
                                         ponents of the colour specification
                                         is out of range.  The valid range is
                                         dependent upon the current colour
                                         model */

/* Transformations and Viewing Errors */
#define PE_MAX_VIEW           (150)   /* Ignoring function, setting this view
                                         table entry would exceed the maximum
                                         number of entries allowed in the
                                         workstations view table */
#define PE_INVALID_WINDOW     (151)   /* Ignoring function, invalid window;
                                         XMIN >= XMAX, YMIN >= YMAX or ZMIN >
                                         ZMAX */
#define PE_INVALID_VIEWPORT   (152)   /* Ignoring function, invalid viewport;
                                         XMIN >= XMAX, YMIN >= YMAX or ZMIN >
                                         ZMAX */
#define PE_INVALID_CLIP       (153)   /* Ignoring function, invalid view clip-
                                         ping limits; XMIN >= XMAX, YMIN >=
                                         YMAX or ZMIN > ZMAX */
#define PE_BAD_CLIP           (154)   /* Ignoring function, the view clipping
                                         limits are not within NPC range */
#define PE_BAD_PROJ_VIEWPORT  (155)   /* Ignoring function, the projection
                                         viewport limits are not withing NPC
                                         range */
#define PE_BAD_WS_WINDOW      (156)   /* Ignoring function, the workstation
                                         window limits are not within NPC
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                        range */
#define PE_BAD_WS_VIEWPORT      (157)   /* Ignoring function, the workstation
                                           viewport is not within display space
                                           */
#define PE_BAD_PLANES           (158)   /* Ignoring function, front plane and
                                           back plane distances are equal when
                                           z-extent of the projection viewport
                                           is zero */
#define PE_BAD_VPN              (159)   /* Ignoring function, the view plane
                                           normal vector has length zero */
#define PE_BAD_VUP              (160)   /* Ignoring function, the view up vector
                                           has length zero */
#define PE_BAD_VUP_VPN          (161)   /* Ignoring function, the view up and
                                           view plane normal vectors are paral-
                                           lel thus the viewing coordinate sys-
                                           tem cannot be established */
#define PE_BAD_PRP              (162)   /* Ignoring function, the projection
                                           reference point is between the front
                                           and back planes */
#define PE_PRP_VIEW_PLANE       (163)   /* Ignoring function, the projection
                                           reference point cannot be positioned
                                           on the view plane */
#define PE_FRONT_BACK           (164)   /* Ignoring function, the back plane is
                                           in front of the front plane */

/* Structure Errors */
#define PE_IGNORE_STRUCTS       (200)   /* Warning, ignoring structures that do
                                           not exist */
#define PE_BAD_STRUCT           (201)   /* Ignoring function, the specified
                                           structure does not exist */
#define PE_BAD_ELEMENT          (202)   /* Ignoring function, the specified ele-
                                           ment does not exist */
#define PE_BAD_PATH             (203)   /* Ignoring function, specified starting
                                           path not found in CSS  */
#define PE_BAD_CEILING_IND      (204)   /* Ignoring function, specified search
                                           ceiling index out of range */
#define PE_NO_LABEL             (205)   /* Ignoring function, the label does not
                                           exist in the open structure between
                                           the element pointer and the end of
                                           the structure */
#define PE_NO_LABELS            (206)   /* Ignoring function, one or both of the
                                           labels does not exist in the open
                                           structure between the element pointer
                                           and the end of the structure */
#define PE_BAD_PATH_DEPTH       (207)   /* Ignoring function, the specified
                                           path depth is less than zero (0) */
#define PE_BAD_DISP_PRI         (208)   /* Ignoring function, the display prior-
                                           ity is out of range */

/* Input Errors */
#define PE_NO_DEVICE            (250)   /* Ignoring function, the specified dev-
                                           ice is not available on the specified
                                           workstation */
#define PE_NOT_REQUEST          (251)   /* Ignoring function, the function
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

```
                                        requires the input device to be in
                                        REQUEST mode */
#define PE_NOT_SAMPLE          (252)    /* Ignoring function, the function
                                        requires the input device to be in
                                        SAMPLE mode */
#define PE_BAD_PET             (253)    /* Warning, the specified prompt/echo
                                        type is not available on the speci-
                                        fied workstation. Prompt/echo type
                                        one will be used in its place */
#define PE_INVALID_ECHO        (254)    /* Ignoring function, invalid echo
                                        area/volume; XMIN >= XMAX, YMIN >=
                                        YMAX or ZMIN > ZMAX */
#define PE_BAD_ECHO            (255)    /* Ignoring function, one of the echo
                                        area/volume boundary points is out-
                                        side the range of the device */
#define PE_QUEUE_OFLOW         (256)    /* Warning, the input queue has over-
                                        flowed */
#define PE_NO_QUEUE_OFLOW      (257)    /* Ignoring function, input queue has
                                        not overflowed */
#define PE_OFLOW_NO_GO         (258)    /* Ignoring function, input queue has
                                        overflowed, but associated worksta-
                                        tion has been closed */
#define PE_BAD_CLASS           (259)    /* Ignoring function, the input device
                                        class of the current input report
                                        does not match the class being
                                        requested */
#define PE_BAD_DATA_REC        (260)    /* Ignoring function, one of the fields
                                        within the input device data record
                                        is in error */
#define PE_INVALID_VALUE       (261)    /* Ignoring function, initial value is
                                        invalid */
#define PE_STROKE_BUF_SIZE     (262)    /* Ignoring function, number of points
                                        in the initial stroke is greater than
                                        the buffer size */
#define PE_STRING_BUF_SIZE     (263)    /* Ignoring function, length of the ini-
                                        tial string is greater than the
                                        buffer size */

/* Metafile Errors */
#define PE_ILLEGAL_ITEM_TYPE   (300)    /* Ignoring function, item type is not
                                        allowed for user items */
#define PE_INVALID_ITEM_LENGTH          (301)/*Ignoring function, item length is
                                        invalid */
#define PE_METAFILE_EMPTY      (302)    /* Ignoring function, no item is left in
                                        metafile input */
#define PE_INVALID_ITEM        (303)    /* Ignoring function, metafile item is
                                        invalid */
#define PE_BAD_ITEM_TYPE       (304)    /* Ignoring function, item type is
                                        unknown */
#define PE_BAD_ITEM_REC        (305)    /* Ignoring function, content of item
                                        data record is invalid for the speci-
                                        fied item type */
#define PE_MAX_ITEM_LENGTH     (306)    /* Ignoring function, maximum item data
                                        record length is invalid */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define PE_USER_ITEM            (307)   /* Ignoring function, user item cannot
                                           be interpreted */

/* Escape Errors */
#define PE_ESCAPE_NOT_AVAIL     (350)   /* Warning, the specified escape is not
                                           available on one or more workstations
                                           in this implementation.  The escape
                                           will be processed by those worksta-
                                           tions on which it is available */
#define PE_BAD_ESCAPE_DATA      (351)   /* Ignoring function, one of the fields
                                           within the escape data record is in
                                           error */

/* Archival and Retrieval Errors */
#define PE_AR_CANT_OPEN         (400)   /* Ignoring function, the archive file
                                           cannot be opened */
#define PE_MAX_AR               (401)   /* Ignoring function, opening this
                                           archive file would exceed the maximum
                                           number of simultaneously open archive
                                           files */
#define PE_DUP_AR_ID            (402)   /* Ignoring function, archive file iden-
                                           tifier already in use */
#define PE_BAD_AR               (403)   /* Ignoring function, the archive file
                                           is not a PHIGS archive file */
#define PE_AR_NOT_OPEN          (404)   /* Ignoring function, the specified
                                           archive file is not open */
#define PE_NAME_CONFLICT        (405)   /* Ignoring function, name conflict
                                           occurred while conflict resolution
                                           flag has value ABANDON */
#define PE_AR_FULL              (406)   /* Warning, the archive file is full.
                                           Any structures that were archived
                                           were archived in total */
#define PE_AR_NO_STRUCT         (407)   /* Warning, some of the specified struc-
                                           tures do not exist on the archive
                                           file */
#define PE_AR_NO_STRUCT_EMPTY   (408)   /* Warning, some of the specified struc-
                                           tures do not exist on the archive
                                           file.  PHIGS will create empty struc-
                                           ture in their places */

/* Miscellaneous Errors */
#define PE_BAD_ERROR_FILE       (450)   /* Ignoring function, the specified
                                           error file is invalid */

/* System Errors */
#define PE_OFLOW_PHIGS          (900)   /* Storage overflow has occurred in
                                           PHIGS */
#define PE_OFLOW_CSS            (901)   /* Storage overflow has occurred in CSS
                                           */
#define PE_IO_ERROR_READ        (902)   /* Input/Output error has occurred while
                                           reading */
#define PE_IO_ERROR_WRITE       (903)   /* Input/Output error has occurred while
                                           writing */
#define PE_IO_ERROR_TO_WS       (904)   /* Input/Output error has occurred while
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                                        sending data to a workstation */
#define PE_IO_ERROR_FROM_WS   (905)   /* Input/Output error has occurred
                                        receiving data from a workstation */
#define PE_IO_ERROR_LIB       (906)   /* Input/Output error has occurred dur-
                                        ing program library management */
#define PE_IO_ERROR_WDT       (907)   /* Input/Output error has occurred while
                                        reading workstation description table
                                        */
#define PE_ARITHMETIC_ERROR   (908)   /* Arithmetic error has occurred */

/*      Binding Specific Errors */
#define PE_START_IND_INVAL    (2200)  /* Ignoring function, start index is out
                                        of range */
#define PE_LIST_LENGTH_LT_ZERO        (2201)/*Ignoring function, the length of
                                        the application's list is negative */
#define PE_ENUM_TYPE_INVAL    (2202)  /* Ignoring function, enumeration type
                                        out of range */
#define PE_ALLOC_STORE        (2203)  /* Ignoring function, error while allo-
                                        cating a Store */
#define PE_ALLOC_STORE_MEM    (2204)  /* Ignoring function, error while allo-
                                        cating memory for a Store */
```

```
#define Pfn_open_phigs                  (0)
#define Pfn_close_phigs                 (1)
#define Pfn_open_ws                     (2)
#define Pfn_close_ws                    (3)
#define Pfn_redraw_all_structs          (4)
#define Pfn_upd_ws                      (5)
#define Pfn_set_disp_upd_st             (6)
#define Pfn_message                     (7)
#define Pfn_polyline3                   (8)
#define Pfn_polyline                    (9)
#define Pfn_polymarker3                 (10)
#define Pfn_polymarker                  (11)
#define Pfn_text3                       (12)
#define Pfn_text                        (13)
#define Pfn_anno_text_rel3              (14)
#define Pfn_anno_text_rel               (15)
#define Pfn_fill_area3                  (16)
#define Pfn_fill_area                   (17)
#define Pfn_fill_area_set3              (18)
#define Pfn_fill_area_set               (19)
#define Pfn_cell_array3                 (20)
#define Pfn_cell_array                  (21)
#define Pfn_gdp3                        (22)
#define Pfn_gdp                         (23)
#define Pfn_set_line_ind                (24)
#define Pfn_set_marker_ind              (25)
#define Pfn_set_text_ind                (26)
#define Pfn_set_int_ind                 (27)
#define Pfn_set_edge_ind                (28)
#define Pfn_set_linetype                (29)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define  Pfn_set_linewidth                (30)
#define  Pfn_set_line_colr_ind            (31)
#define  Pfn_set_marker_type              (32)
#define  Pfn_set_marker_size              (33)
#define  Pfn_set_marker_colr_ind          (34)
#define  Pfn_set_text_font                (35)
#define  Pfn_set_text_prec                (36)
#define  Pfn_set_char_expan               (37)
#define  Pfn_set_char_space               (38)
#define  Pfn_set_text_colr_ind            (39)
#define  Pfn_set_char_ht                  (40)
#define  Pfn_set_char_up_vec              (41)
#define  Pfn_set_text_path                (42)
#define  Pfn_set_text_align               (43)
#define  Pfn_set_anno_char_ht             (44)
#define  Pfn_set_anno_char_up_vec         (45)
#define  Pfn_set_anno_path                (46)
#define  Pfn_set_anno_align               (47)
#define  Pfn_set_anno_style               (48)
#define  Pfn_set_int_style                (49)
#define  Pfn_set_int_style_ind            (50)
#define  Pfn_set_int_colr_ind             (51)
#define  Pfn_set_edge_flag                (52)
#define  Pfn_set_edgetype                 (53)
#define  Pfn_set_edgewidth                (54)
#define  Pfn_set_edge_colr_ind            (55)
#define  Pfn_set_pat_size                 (56)
#define  Pfn_set_pat_ref_point_vecs       (57)
#define  Pfn_set_pat_ref_point            (58)
#define  Pfn_add_names_set                (59)
#define  Pfn_remove_names_set             (60)
#define  Pfn_set_indiv_asf                (61)
#define  Pfn_set_line_rep                 (62)
#define  Pfn_set_marker_rep               (63)
#define  Pfn_set_text_rep                 (64)
#define  Pfn_set_int_rep                  (65)
#define  Pfn_set_edge_rep                 (66)
#define  Pfn_set_pat_rep                  (67)
#define  Pfn_set_colr_rep                 (68)
#define  Pfn_set_highl_filter             (69)
#define  Pfn_set_invis_filter             (70)
#define  Pfn_set_colr_model               (71)
#define  Pfn_set_hlhsr_id                 (72)
#define  Pfn_set_hlhsr_mode               (73)
#define  Pfn_set_local_tran3              (74)
#define  Pfn_set_local_tran               (75)
#define  Pfn_set_global_tran3             (76)
#define  Pfn_set_global_tran              (77)
#define  Pfn_set_model_clip_vol3          (78)
#define  Pfn_set_model_clip_vol           (79)
#define  Pfn_set_model_clip_ind           (80)
#define  Pfn_restore_model_clip_vol       (81)
#define  Pfn_set_view_ind                 (82)
#define  Pfn_set_view_rep3                (83)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577497 817 ■


```
#define  Pfn_set_view_rep                  (84)
#define  Pfn_set_view_tran_in_pri          (85)
#define  Pfn_set_ws_win3                   (86)
#define  Pfn_set_ws_win                    (87)
#define  Pfn_set_ws_vp3                    (88)
#define  Pfn_set_ws_vp                     (89)
#define  Pfn_open_struct                   (90)
#define  Pfn_close_struct                  (91)
#define  Pfn_exec_struct                   (92)
#define  Pfn_label                         (93)
#define  Pfn_appl_data                     (94)
#define  Pfn_gse                           (95)
#define  Pfn_set_edit_mode                 (96)
#define  Pfn_copy_all_elems_struct         (97)
#define  Pfn_set_elem_ptr                  (98)
#define  Pfn_offset_elem_ptr               (99)
#define  Pfn_set_elem_ptr_label            (100)
#define  Pfn_del_elem                      (101)
#define  Pfn_del_elem_range                (102)
#define  Pfn_del_elems_labels              (103)
#define  Pfn_empty_struct                  (104)
#define  Pfn_del_struct                    (105)
#define  Pfn_del_struct_net                (106)
#define  Pfn_del_all_structs               (107)
#define  Pfn_change_struct_id              (108)
#define  Pfn_change_struct_refs            (109)
#define  Pfn_change_struct_id_refs         (110)
#define  Pfn_post_struct                   (111)
#define  Pfn_unpost_struct                 (112)
#define  Pfn_unpost_all_structs            (113)
#define  Pfn_open_ar_file                  (114)
#define  Pfn_close_ar_file                 (115)
#define  Pfn_ar_structs                    (116)
#define  Pfn_ar_struct_nets                (117)
#define  Pfn_ar_all_structs                (118)
#define  Pfn_set_conf_res                  (119)
#define  Pfn_ret_struct_ids                (120)
#define  Pfn_ret_structs                   (121)
#define  Pfn_ret_struct_nets               (122)
#define  Pfn_ret_all_structs               (123)
#define  Pfn_ret_paths_ances               (124)
#define  Pfn_ret_paths_descs               (125)
#define  Pfn_del_structs_ar                (126)
#define  Pfn_del_struct_nets_ar            (127)
#define  Pfn_del_all_structs_ar            (128)
#define  Pfn_set_pick_id                   (129)
#define  Pfn_set_pick_filter               (130)
#define  Pfn_init_loc3                     (131)
#define  Pfn_init_loc                      (132)
#define  Pfn_init_stroke3                  (133)
#define  Pfn_init_stroke                   (134)
#define  Pfn_init_val3                     (135)
#define  Pfn_init_val                      (136)
#define  Pfn_init_choice3                  (137)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577498 753 ■


```
#define  Pfn_init_choice                (138)
#define  Pfn_init_pick3                 (139)
#define  Pfn_init_pick                  (140)
#define  Pfn_init_string3               (141)
#define  Pfn_init_string                (142)
#define  Pfn_set_loc_mode               (143)
#define  Pfn_set_stroke_mode            (144)
#define  Pfn_set_val_mode               (145)
#define  Pfn_set_choice_mode            (146)
#define  Pfn_set_pick_mode              (147)
#define  Pfn_set_string_mode            (148)
#define  Pfn_req_loc3                   (149)
#define  Pfn_req_loc                    (150)
#define  Pfn_req_stroke3                (151)
#define  Pfn_req_stroke                 (152)
#define  Pfn_req_val                    (153)
#define  Pfn_req_choice                 (154)
#define  Pfn_req_pick                   (155)
#define  Pfn_req_string                 (156)
#define  Pfn_sample_loc3                (157)
#define  Pfn_sample_loc                 (158)
#define  Pfn_sample_stroke3             (159)
#define  Pfn_sample_stroke              (160)
#define  Pfn_sample_val                 (161)
#define  Pfn_sample_choice              (162)
#define  Pfn_sample_pick                (163)
#define  Pfn_sample_string              (164)
#define  Pfn_await_event                (165)
#define  Pfn_flush_events               (166)
#define  Pfn_get_loc3                   (167)
#define  Pfn_get_loc                    (168)
#define  Pfn_get_stroke3                (169)
#define  Pfn_get_stroke                 (170)
#define  Pfn_get_val                    (171)
#define  Pfn_get_choice                 (172)
#define  Pfn_get_pick                   (173)
#define  Pfn_get_string                 (174)
#define  Pfn_write_item                 (175)
#define  Pfn_get_item_type              (176)
#define  Pfn_read_item                  (177)
#define  Pfn_interpret_item             (178)
#define  Pfn_set_err_hand_mode          (179)
#define  Pfn_escape                     (180)
#define  Pfn_set_err_hand               (181)


#define  PLINE_SOLID                    (1)
#define  PLINE_DASH                     (2)
#define  PLINE_DOT                      (3)
#define  PLINE_DASH_DOT                 (4)



#define  PMARKER_DOT                    (1)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define  PMARKER_PLUS                    (2)
#define  PMARKER_ASTERISK                (3)
#define  PMARKER_CIRCLE                  (4)
#define  PMARKER_CROSS                   (5)


#define PANNO_STYLE_UNCONNECTED          (1)
#define PANNO_STYLE_LEAD_LINE            (2)


#define  PMODEL_RGB                      (1)
#define  PMODEL_CIELUV                   (2)
#define  PMODEL_HSV                      (3)
#define  PMODEL_HLS                      (4)


#define  PLOC_DEF                        (1)
#define  PLOC_CROSS_HAIR                 (2)
#define  PLOC_TRACK_CROSS                (3)
#define  PLOC_RUB_BAND                   (4)
#define  PLOC_RECT                       (5)
#define  PLOC_DIGIT                      (6)

#define  PSTROKE_DEF                     (1)
#define  PSTROKE_DIGIT                   (2)
#define  PSTROKE_MARKER                  (3)
#define  PSTROKE_LINE                    (4)

#define  PVAL_DEF                        (1)
#define  PVAL_GRAPH                      (2)
#define  PVAL_DIGIT                      (3)

#define  PCHOICE_DEF                     (1)
#define  PCHOICE_PR_ECHO                 (2)
#define  PCHOICE_STRING_PR               (3)
#define  PCHOICE_STRING_IN               (4)
#define  PCHOICE_STRUCT                  (5)

#define  PPICK_DEF                       (1)
#define  PPICK_GROUP_HIGHL               (2)
#define  PPICK_STRUCT_NETWORK            (3)

#define PSTRING_DEF                      (1)


#define  PDEF_MEM_SIZE                   ((size_t) (-1))
#define  PDEF_ERR_FILE                   ((char *) (0))


#define  PFIRST_PHIGS_ELEM               (PELEM_POLYLINE3)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577500 131 ■


```
#define  PLAST_PHIGS_ELEM                    (PELEM_PICK_ID)
```

A.2 Types in compilation order

```
typedef float    Pfloat;
typedef int  Pint;


typedef Pfloat Pmatrix3[4][4];


typedef Pfloat Pmatrix[3][3];


typedef void *Pstore;


typedef enum {
    PWS_INDEP,
    PWS_DEP
} Pws_dep_ind;


typedef enum {
    PSYS_ST_PHCL,
    PSYS_ST_PHOP
} Psys_st;


typedef enum {
    PWS_ST_WSCL,
    PWS_ST_WSOP
} Pws_st;


typedef enum {
    PSTRUCT_ST_STCL,
    PSTRUCT_ST_STOP
} Pstruct_st;


typedef enum {
    PSTRUCT_STATUS_NON_EXISTENT,
    PSTRUCT_STATUS_EMPTY,
    PSTRUCT_STATUS_NOT_EMPTY
} Pstruct_status;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
    PST_ARCL,
    PST_AROP
} Par_st;


typedef enum {
    PCLASS_VEC,
    PCLASS_RASTER,
    PCLASS_OTHER
} Pws_class;


typedef enum {
    PCAT_OUT,
    PCAT_IN,
    PCAT_OUTIN,
    PCAT_MO,
    PCAT_MI
} Pws_cat;


typedef enum {
    PFLAG_COND,
    PFLAG_ALWAYS
} Pctrl_flag;


typedef enum {
    PFLAG_POSTPONE,
    PFLAG_PERFORM
} Pregen_flag;


typedef enum {
    PDEFER_ASAP,
    PDEFER_BNIG,
    PDEFER_BNIL,
    PDEFER_ASTI,
    PDEFER_WAIT
} Pdefer_mode;


typedef enum {
    PMODE_NIVE,
    PMODE_UWOR,
    PMODE_UQUM
} Pmod_mode;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577502 T04 ■


```
typedef enum {
    PSIMULT_NO_MORE,
    PSIMULT_MORE
} Pmore_simult_events;


typedef enum {
    PNET_CSS,
    PNET_AR
} Pstruct_net_source;


typedef enum {
    PSURF_NOT_EMPTY,
    PSURF_EMPTY
} Pdisp_surf_empty;


typedef enum {
    PVISUAL_ST_CORRECT,
    PVISUAL_ST_DEFER,
    PVISUAL_ST_SIMULATED
} Pvisual_st;


typedef enum {
    PPREC_STRING,
    PPREC_CHAR,
    PPREC_STROKE
} Ptext_prec;


typedef enum {
    PPATH_RIGHT,
    PPATH_LEFT,
    PPATH_UP,
    PPATH_DOWN
} Ptext_path;


typedef enum {
    PHOR_NORM,
    PHOR_LEFT,
    PHOR_CTR,
    PHOR_RIGHT
} Phor_text_align;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef enum {
    PVERT_NORM,
    PVERT_TOP,
    PVERT_CAP,
    PVERT_HALF,
    PVERT_BASE,
    PVERT_BOTTOM
} Pvert_text_align;


typedef enum {
    PSTYLE_HOLLOW,
    PSTYLE_SOLID,
    PSTYLE_PAT,
    PSTYLE_HATCH,
    PSTYLE_EMPTY
} Pint_style;


typedef enum {
    PEDGE_OFF,
    PEDGE_ON
} Pedge_flag;


typedef enum {
    PASPECT_LINETYPE,
    PASPECT_LINEWIDTH,
    PASPECT_LINE_COLR_IND,
    PASPECT_MARKER_TYPE,
    PASPECT_MARKER_SIZE,
    PASPECT_MARKER_COLR_IND,
    PASPECT_TEXT_FONT,
    PASPECT_TEXT_PREC,
    PASPECT_CHAR_EXPAN,
    PASPECT_CHAR_SPACE,
    PASPECT_TEXT_COLR_IND,
    PASPECT_INT_STYLE,
    PASPECT_INT_STYLE_IND,
    PASPECT_INT_COLR_IND,
    PASPECT_EDGE_FLAG,
    PASPECT_EDGETYPE,
    PASPECT_EDGEWIDTH,
    PASPECT_EDGE_COLR_IND
} Paspect;


typedef enum {
    PASF_BUNDLED,
    PASF_INDIV
} Pasf;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577504 887 ■


```
typedef enum {
    PAVAIL_MONOCHR,
    PAVAIL_COLR
} Pcolr_avail;


typedef enum {
    PTYPE_PRECONCAT,
    PTYPE_POSTCONCAT,
    PTYPE_REPLACE
} Pcompose_type;


typedef enum {
    PTYPE_PARAL,
    PTYPE_PERSPECT
} Pproj_type;


typedef enum {
    PIND_NO_CLIP,
    PIND_CLIP
} Pclip_ind;


typedef enum {
    PPRI_HIGHER,
    PPRI_LOWER
} Prel_pri;


typedef enum {
    PRES_MAINTAIN,
    PRES_ABANDON,
    PRES_UPD
} Pconf_res;


typedef enum {
    PFLAG_LINE,
    PFLAG_FILL,
    PFLAG_FILL_SET
} Pline_fill_ctrl_flag;


typedef enum {
    PORDER_TOP_FIRST,
    PORDER_BOTTOM_FIRST
} Ppath_order;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef enum {
    POP_REQ,
    POP_SAMPLE,
    POP_EVENT
} Pop_mode;


typedef enum {
    PSWITCH_NO_ECHO,
    PSWITCH_ECHO
} Pecho_switch;


typedef enum {
    PIN_STATUS_NONE,
    PIN_STATUS_OK,
    PIN_STATUS_NO_IN
} Pin_status;


typedef enum {
    PSTRUCT_NONE,
    PSTRUCT_OPEN
} Popen_struct_status;


typedef enum {
    PIN_NONE,
    PIN_LOC,
    PIN_STROKE,
    PIN_VAL,
    PIN_CHOICE,
    PIN_PICK,
    PIN_STRING
} Pin_class;


typedef enum {
    PPR_OFF,
    PPR_ON
} Ppr_switch;


typedef enum {
    PINQ_SET,
    PINQ_REALIZED
} Pinq_type;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef enum {
    PUPD_NOT_PEND,
    PUPD_PEND
} Pupd_st;


typedef enum {
    PDC_METRES,
    PDC_OTHER
} Pdc_units;


typedef enum {
    PDYN_IRG,
    PDYN_IMM,
    PDYN_CBS
} Pdyn_mod;


typedef enum {
    PATTR_LINE,
    PATTR_MARKER,
    PATTR_TEXT,
    PATTR_INT,
    PATTR_EDGE
} Pattrs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef enum {
    PELEM_ALL,
    PELEM_NIL,
    PELEM_POLYLINE3,
    PELEM_POLYLINE,
    PELEM_POLYMARKER3,
    PELEM_POLYMARKER,
    PELEM_TEXT3,
    PELEM_TEXT,
    PELEM_ANNO_TEXT_REL3,
    PELEM_ANNO_TEXT_REL,
    PELEM_FILL_AREA3,
    PELEM_FILL_AREA,
    PELEM_FILL_AREA_SET3,
    PELEM_FILL_AREA_SET,
    PELEM_CELL_ARRAY3,
    PELEM_CELL_ARRAY,
    PELEM_GDP3,
    PELEM_GDP,
    PELEM_LINE_IND,
    PELEM_MARKER_IND,
    PELEM_TEXT_IND,
    PELEM_INT_IND,
    PELEM_EDGE_IND,
    PELEM_LINETYPE,
    PELEM_LINEWIDTH,
    PELEM_LINE_COLR_IND,
    PELEM_MARKER_TYPE,
    PELEM_MARKER_SIZE,
    PELEM_MARKER_COLR_IND,
    PELEM_TEXT_FONT,
    PELEM_TEXT_PREC,
    PELEM_CHAR_EXPAN,
    PELEM_CHAR_SPACE,
    PELEM_TEXT_COLR_IND,
    PELEM_CHAR_HT,
    PELEM_CHAR_UP_VEC,
    PELEM_TEXT_PATH,
    PELEM_TEXT_ALIGN,
    PELEM_ANNO_CHAR_HT,
    PELEM_ANNO_CHAR_UP_VEC,
    PELEM_ANNO_PATH,
    PELEM_ANNO_ALIGN,
    PELEM_ANNO_STYLE,
    PELEM_INT_STYLE,
    PELEM_INT_STYLE_IND,
    PELEM_INT_COLR_IND,
    PELEM_EDGE_FLAG,
    PELEM_EDGETYPE,
    PELEM_EDGEWIDTH,
    PELEM_EDGE_COLR_IND,
    PELEM_PAT_SIZE,
    PELEM_PAT_REF_POINT_VECS,
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    PELEM_PAT_REF_POINT,
    PELEM_ADD_NAMES_SET,
    PELEM_REMOVE_NAMES_SET,
    PELEM_INDIV_ASF,
    PELEM_HLHSR_ID,
    PELEM_LOCAL_MODEL_TRAN3,
    PELEM_LOCAL_MODEL_TRAN,
    PELEM_GLOBAL_MODEL_TRAN3,
    PELEM_GLOBAL_MODEL_TRAN,
    PELEM_MODEL_CLIP_VOL3,
    PELEM_MODEL_CLIP_VOL,
    PELEM_MODEL_CLIP_IND,
    PELEM_RESTORE_MODEL_CLIP_VOL,
    PELEM_VIEW_IND,
    PELEM_EXEC_STRUCT,
    PELEM_LABEL,
    PELEM_APPL_DATA,
    PELEM_GSE,
    PELEM_PICK_ID
} Pelem_type;


typedef enum {
    PEDIT_INSERT,
    PEDIT_REPLACE
} Pedit_mode;


typedef enum {
    PFLAG_DEL,
    PFLAG_KEEP
} Pref_flag;


typedef enum {
    PERR_OFF,
    PERR_ON
} Perr_mode;


typedef enum {
    PDIR_BACKWARD,
    PDIR_FORWARD
} Psearch_dir;


typedef enum {
    PSEARCH_STATUS_FAILURE,
    PSEARCH_STATUS_SUCCESS
} Psearch_status;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  size_t  size;       /* sizeof data     */
  void    *data;      /* pointer to data */
} Pdata;


typedef struct {
  Pfloat  cieluv_x;          /* x coefficient */
  Pfloat  cieluv_y;          /* y coefficient */
  Pfloat  cieluv_y_lum;      /* y luminance   */
} Pcieluv;


typedef struct {
  Pfloat  hue;            /* hue        */
  Pfloat  lightness;      /* lightness  */
  Pfloat  satur;          /* saturation */
} Phls;


typedef struct {
  Pfloat  hue;        /* hue        */
  Pfloat  satur;      /* saturation */
  Pfloat  value;      /* value      */
} Phsv;


typedef struct {
  Pfloat  red;        /* red intensity   */
  Pfloat  green;      /* green intensity */
  Pfloat  blue;       /* blue intensity  */
} Prgb;


typedef union {
  Prgb           rgb;        /* Red Green Blue colour specification           */
  Pcieluv        cieluv;     /* CIE L*u*v* colour specification               */
  Phls           hls;        /* Hue Lightness Saturation colour specification */
  Phsv           hsv;        /* Hue Saturation Value colour specification     */
  Pdata          unsupp;     /* colour in an unsupported colour model         */
  int impl_dep;              /* implementation defined                        */
} Pcolr_rep;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint  num_ints;      /* number of Pints in list */
  Pint  *ints;         /* list of integers        */
} Pint_list;


typedef struct {
  Pint         id;       /* GSE identifier                                */
  Pws_dep_ind  ind;      /* workstation independent/dependent indicator */
} Pgse_id_dep;


typedef struct {
  Pint         num_id_facs;      /* number of identifiers/dependency element */
  Pgse_id_dep  *id_facs;         /* list of GSE facilities                   */
} Pgse_id_dep_list;


typedef struct {
  Pint   id;         /* archive file identifier */
  char   *name;      /* archive file name       */
} Par_file;


typedef struct {
  Pint      num_ar_files;      /* number of archive files */
  Par_file  *ar_files;         /* list of archive files   */
} Par_file_list;


typedef struct {
  Pint  struct_id;      /* structure identifier */
  Pint  elem_pos;       /* element position     */
} Pelem_ref;


typedef struct {
  Pint       num_elem_refs;      /* number of element references */
  Pelem_ref  *elem_refs;         /* list of element references   */
} Pelem_ref_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
typedef struct {
  Pint           num_elem_ref_lists;     /* number of element references lists */
  Pelem_ref_list *elem_ref_lists;        /* list of element reference lists    */
} Pelem_ref_list_list;


typedef struct {
  Pint    id;           /* structure id     */
  Pfloat  disp_pri;     /* display priority */
} Pposted_struct;


typedef struct {
  Pint           num_postings;     /* number of structure postings */
  Pposted_struct *postings;        /* list of postings             */
} Pposted_struct_list;


typedef struct {
  Pfloat  x;     /* x coordinate */
  Pfloat  y;     /* y coordinate */
} Ppoint;


typedef struct {
  Pfloat  x;     /* x coordinate */
  Pfloat  y;     /* y coordinate */
  Pfloat  z;     /* z coordinate */
} Ppoint3;


typedef struct {
  Pfloat  delta_x;     /* delta x value */
  Pfloat  delta_y;     /* delta y value */
} Pvec;


typedef struct {
  Pfloat  delta_x;     /* delta x value */
  Pfloat  delta_y;     /* delta y value */
  Pfloat  delta_z;     /* delta z value */
} Pvec3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint  size_x;      /* x size */
  Pint  size_y;      /* y size */
} Pint_size;

typedef struct {
  Pint  size_x;      /* x size */
  Pint  size_y;      /* y size */
  Pint  size_z;      /* z size */
} Pint_size3;

typedef struct {
  Pfloat  size_x;      /* x size */
  Pfloat  size_y;      /* y size */
} Pfloat_size;

typedef struct {
  Pfloat  size_x;      /* x size */
  Pfloat  size_y;      /* y size */
  Pfloat  size_z;      /* z size */
} Pfloat_size3;

typedef struct {
  Ppoint  point;      /* point  */
  Pvec    norm;       /* normal */
} Phalf_space;

typedef struct {
  Ppoint3  point;      /* point  */
  Pvec3    norm;       /* normal */
} Phalf_space3;

typedef struct {
  Ppoint  p;      /* point p */
  Ppoint  q;      /* point q */
} Prect;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577513 89T ■


```
typedef struct {
  Ppoint3  p;      /* point p */
  Ppoint3  q;      /* point q */
  Ppoint3  r;      /* point r */
} Pparal;


typedef struct {
  Pfloat  x_min;      /* x min */
  Pfloat  x_max;      /* x max */
  Pfloat  y_min;      /* y min */
  Pfloat  y_max;      /* y max */
  Pfloat  z_min;      /* z min */
  Pfloat  z_max;      /* z max */
} Plimit3;


typedef struct {
  Pfloat  x_min;      /* x min */
  Pfloat  x_max;      /* x max */
  Pfloat  y_min;      /* y min */
  Pfloat  y_max;      /* y max */
} Plimit;


typedef struct {
  Pint    num_points;      /* number of Ppoints in the list */
  Ppoint  *points;         /* list of points                */
} Ppoint_list;


typedef struct {
  Pint     num_points;      /* number of Ppoint3s in the list */
  Ppoint3  *points;         /* list of points                 */
} Ppoint_list3;


typedef struct {
  Pint         num_point_lists;      /* number of point lists */
  Ppoint_list  *point_lists;         /* list of point lists   */
} Ppoint_list_list;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint          num_point_lists;     /* number of point lists */
  Ppoint_list3  *point_lists;        /* list of point lists   */
} Ppoint_list_list3;

typedef struct {
  Pint         num_half_spaces;     /* number of half spaces */
  Phalf_space  *half_spaces;        /* list of half spaces   */
} Phalf_space_list;

typedef struct {
  Pint          num_half_spaces;     /* number of half spaces */
  Phalf_space3  *half_spaces;        /* list of half spaces   */
} Phalf_space_list3;

typedef struct {
  Pint_list  incl_set;     /* inclusion set */
  Pint_list  excl_set;     /* exclusion set */
} Pfilter;

typedef struct {
  Pint     num_filters;     /* number of filters */
  Pfilter  *filters;        /* list of filters   */
} Pfilter_list;

typedef struct {
  Phor_text_align   hor;      /* horizontal component */
  Pvert_text_align  vert;     /* vertical component   */
} Ptext_align;

typedef struct {
  Pint    type;          /* linetype                */
  Pfloat  width;         /* linewidth scale factor */
  Pint    colr_ind;      /* colour index            */
} Pline_bundle;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint    type;          /* marker type                */
  Pfloat  size;          /* marker size scale factor */
  Pint    colr_ind;      /* colour index               */
} Pmarker_bundle;


typedef struct {
  Pint        font;           /* text font                    */
  Ptext_prec  prec;           /* text precision               */
  Pfloat      char_expan;     /* character expansion factor */
  Pfloat      char_space;     /* character spacing            */
  Pint        colr_ind;       /* text colour index            */
} Ptext_bundle;


typedef struct {
  Pint_style  style;          /* interior style        */
  Pint        style_ind;      /* interior style index  */
  Pint        colr_ind;       /* interior colour index */
} Pint_bundle;


typedef struct {
  Pedge_flag  flag;           /* edge flag              */
  Pint        type;           /* edgetype               */
  Pfloat      width;          /* edgewidth scale factor */
  Pint        colr_ind;       /* edge colour index      */
} Pedge_bundle;


typedef struct {
  Pint_size  dims;             /* pattern's dimensions */
  Pint       *colr_array;      /* colour index array   */
} Ppat_rep;


typedef struct {
  Pmatrix3   ori_matrix;      /* orientation matrix       */
  Pmatrix3   map_matrix;      /* mapping matrix           */
  Plimit3    clip_limit;      /* clipping limits          */
  Pclip_ind  xy_clip;         /* X-Y clipping indicator   */
  Pclip_ind  back_clip;       /* back clipping indicator  */
  Pclip_ind  front_clip;      /* front clipping indicator */
} Pview_rep3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pmatrix    ori_matrix;      /* orientation matrix     */
  Pmatrix    map_matrix;      /* mapping matrix         */
  Plimit     clip_limit;      /* clipping limits        */
  Pclip_ind  xy_clip;         /* X-Y clipping indicator */
} Pview_rep;


typedef struct {
  Plimit      win;                /* window limits              */
  Plimit3     proj_vp;            /* projection viewport limits */
  Pproj_type  proj_type;          /* projection type            */
  Ppoint3     proj_ref_point;     /* projection reference point */
  Pfloat      view_plane;         /* view plane distance        */
  Pfloat      back_plane;         /* back plane distance        */
  Pfloat      front_plane;        /* front plane distance       */
} Pview_map3;


typedef struct {
  Plimit  win;          /* window limits              */
  Plimit  proj_vp;      /* projection viewport limits */
} Pview_map;


typedef struct {
  Pasf          type_asf;         /* linetype asf          */
  Pasf          width_asf;        /* linewidth asf         */
  Pasf          colr_ind_asf;     /* line colour index asf */
  Pint          ind;              /* line index            */
  Pline_bundle  bundle;           /* line bundle           */
} Pline_attrs;


typedef struct {
  Pasf         style_asf;         /* interior style asf        */
  Pasf         style_ind_asf;     /* interior style index asf  */
  Pasf         colr_ind_asf;      /* interior colour index asf */
  Pint         ind;               /* interior index            */
  Pint_bundle  bundle;            /* interior bundle           */
} Pint_attrs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pasf          flag_asf;         /* edge flag asf         */
  Pasf          type_asf;         /* edgetype asf          */
  Pasf          width_asf;        /* edgewidth asf         */
  Pasf          colr_ind_asf;     /* edge colour index asf */
  Pint          ind;              /* edge index            */
  Pedge_bundle  bundle;           /* edge bundle           */
} Pedge_attrs;


typedef struct {
  Pasf            type_asf;         /* marker type asf              */
  Pasf            size_asf;         /* marker size scale factor asf */
  Pasf            colr_ind_asf;     /* marker colour index asf      */
  Pint            ind;              /* marker index                 */
  Pmarker_bundle  bundle;           /* marker bundle                */
} Pmarker_attrs;


typedef struct {
  Pint  struct_id;     /* structure identifier */
  Pint  pick_id;       /* pick identifier      */
  Pint  elem_pos;      /* element position     */
} Ppick_path_elem;


typedef struct {
  Pint             depth;          /* pick path depth */
  Ppick_path_elem  *path_list;     /* pick path list  */
} Ppick_path;


typedef struct {
  Pdc_units    dc_units;        /* device coordinate units         */
  Pfloat_size  size_dc;         /* device size in coordinate units */
  Pint_size    size_raster;     /* device size in raster units     */
} Pdisp_space_size;


typedef struct {
  Pdc_units     dc_units;         /* device coordinate units           */
  Pfloat_size3  size_dc;          /* device volume in coordinate units */
  Pint_size3    size_raster;      /* device volume in raster units     */
} Pdisp_space_size3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pdyn_mod  line_bundle;       /* polyline bundle representation   */
  Pdyn_mod  marker_bundle;     /* polymarker bundle representation */
  Pdyn_mod  text_bundle;       /* text bundle representation       */
  Pdyn_mod  int_bundle;        /* interior bundle representation   */
  Pdyn_mod  edge_bundle;       /* edge bundle representation       */
  Pdyn_mod  pat_rep;           /* pattern representation           */
  Pdyn_mod  colr_rep;          /* colour representation            */
  Pdyn_mod  view_rep;          /* view representation              */
  Pdyn_mod  ws_tran;           /* workstation transformation       */
  Pdyn_mod  highl_filter;      /* highlighting filter              */
  Pdyn_mod  invis_filter;      /* invisibility filter              */
  Pdyn_mod  hlhsr_mode;        /* HLHSR mode                       */
} Pdyns_ws_attrs;

typedef struct {
  Pint_list  types;             /* list of linetypes                    */
  Pint       num_widths;        /* number of available linewidths       */
  Pfloat     nom_width;         /* nominal linewidth                    */
  Pfloat     min_width;         /* min linewidth                        */
  Pfloat     max_width;         /* max linewidth                        */
  Pint       num_pred_inds;     /* number of predefined bundle indices */
} Pline_facs;

typedef struct {
  Pint_list  types;             /* list of marker types                 */
  Pint       num_sizes;         /* number of available marker sizes     */
  Pfloat     nom_size;          /* nominal marker size                  */
  Pfloat     min_size;          /* min marker size                      */
  Pfloat     max_size;          /* max marker size                      */
  Pint       num_pred_inds;     /* number of predefined bundle indices */
} Pmarker_facs;

typedef struct {
  Pint        font;      /* text font      */
  Ptext_prec  prec;      /* text precision */
} Ptext_font_prec;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint             num_font_precs;      /* number of fonts and precisions        */
  Ptext_font_prec  *font_precs;         /* list of fonts and precisions          */
  Pint             num_char_hts;        /* number of character heights           */
  Pfloat           min_char_ht;         /* minimum height                        */
  Pfloat           max_char_ht;         /* maximum height                        */
  Pint             num_char_expans;     /* number of character expansion factors */
  Pfloat           min_char_expan;      /* minimum expansion factor              */
  Pfloat           max_char_expan;      /* maximum expansion factor              */
  Pint             num_pred_inds;       /* number of predefined bundle indices   */
} Ptext_facs;

typedef struct {
  Pint        num_int_styles;     /* number of interior styles           */
  Pint_style  int_styles[5];      /* list of available interior styles   */
  Pint_list   hatch_styles;       /* list of available hatch styles      */
  Pint        num_pred_inds;      /* number of predefined bundle indices */
} Pint_facs;

typedef struct {
  Pint_list  types;              /* list of edgetypes                    */
  Pint       num_widths;         /* number of available edgewidths       */
  Pfloat     nom_width;          /* nominal edgewidth                    */
  Pfloat     min_width;          /* min edgewidth                        */
  Pfloat     max_width;          /* max edgewidth                        */
  Pint       num_pred_inds;      /* number of predefined bundle indices */
} Pedge_facs;

typedef struct {
  Pint         num_colrs;          /* number of colours                   */
  Pcolr_avail  colr_avail;         /* colour availability                 */
  Pint         num_pred_inds;      /* number of predefined colour indices */
  Pcieluv      prim_colrs[3];      /* primary colours                     */
} Pcolr_facs;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint  line_bundles;      /* max. # of polyline table entries   */
  Pint  mark_bundles;      /* max. # of polymarker table entries */
  Pint  text_bundles;      /* max. # of text table entries       */
  Pint  int_bundles;       /* max. # of interior table entries   */
  Pint  edge_bundles;      /* max. # of edge table entries       */
  Pint  pat_reps;          /* max. # of pattern table entries    */
  Pint  colr_reps;         /* max. # of colour table entries     */
  Pint  view_reps;         /* max. # of view table entries       */
} Pws_st_tables;


typedef struct {
  Pdyn_mod  content;      /* structure content    */
  Pdyn_mod  post;         /* post structure       */
  Pdyn_mod  unpost;       /* unpost structure     */
  Pdyn_mod  del;          /* delete structure     */
  Pdyn_mod  ref;          /* structure reference  */
} Pdyns_structs;


typedef struct {
  Pint  loc;          /* locators  */
  Pint  stroke;       /* strokes   */
  Pint  val;          /* valuators */
  Pint  choice;       /* choices   */
  Pint  pick;         /* picks     */
  Pint  string;       /* strings   */
} Pnum_in;


typedef struct {
  Pint        num_elem_types;      /* number of element types */
  Pelem_type  *elem_types;         /* list of element types   */
} Pelem_type_list;


typedef union {
  int impl_dep;              /* Metafile Records                */
  Pdata          unsupp;     /* unsupported Metafile item data */
} Pitem_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577521 966 ■


```
typedef struct {
  union Ploc_pets {
    struct Ploc_pet_r1 {
                int impl_dep;                                   /* impl. dependent       */
    } pet_r1;
    struct Ploc_pet_r2 {
                int impl_dep;                                   /* impl. dependent       */
    } pet_r2;
    struct Ploc_pet_r3 {
                int impl_dep;                                   /* impl. dependent       */
    } pet_r3;
    struct Ploc_pet_r4 {
                Pline_attrs          line_attrs;                /* polyline attributes   */
                int impl_dep;                                   /* impl. dependent       */
    } pet_r4;
    struct Ploc_pet_r5 {
                Pline_fill_ctrl_flag line_fill_ctrl_flag;       /* control flag          */
                union Ploc_attrs {
                                     Pline_attrs                line_attrs;
                                                                /* polyline attributes   */
                                     Pint_attrs                 int_attrs;
                                                                /* interior attributes   */
                                     struct Ploc_fill_set {
                                                                Pint_attrs  int_attrs;
                                                                /* interior attributes   */
                                                                Pedge_attrs  edge_attrs;
                                                                /* edge attributes       */
                                     } fill_set;
                } attrs;
                int impl_dep;                                   /* impl. dependent       */
    } pet_r5;
    struct Ploc_pet_r6 {
                int impl_dep;                                   /* impl. dependent       */
    } pet_r6;
    int impl_dep;                                               /* impl. defined PET's   */
  } pets;
} Ploc_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint    in_buf_size;                                  /* input buffer size          */
  Pint    init_pos;                                     /* initial editing position   */
  Pfloat  x_interval;                                   /* x interval                 */
  Pfloat  y_interval;                                   /* y interval                 */
  Pfloat  time_interval;                                /* time interval              */
  union Pstroke_pets {
          struct Pstroke_pet_r1 {
                        int impl_dep;                   /* implementation dependent   */
          } pet_r1;
          struct Pstroke_pet_r2 {
                        int impl_dep;                   /* implementation dependent   */
          } pet_r2;
          struct Pstroke_pet_r3 {
                        Pmarker_attrs  marker_attrs;    /* marker attributes          */
                        int impl_dep;                   /* implementation dependent   */
          } pet_r3;
          struct Pstroke_pet_r4 {
                        Pline_attrs    line_attrs;      /* line attributes            */
                        int impl_dep;                   /* implementation dependent   */
          } pet_r4;
          int impl_dep;                                 /* implementation defined PET's */
  } pets;
} Pstroke_data;


typedef struct {
  Pfloat  low_value;                    /* low value of valuator range  */
  Pfloat  high_value;                   /* high value of valuator range */
  union Pval_pets {
          struct Pval_pet_r1 {
                        int impl_dep;   /* implementation dependent     */
          } pet_r1;
          int impl_dep;                 /* implementation defined PET's */
  } pets;
} Pval_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  union Pchoice_pets {
    struct Pchoice_pet_r1 {
                int impl_dep;                   /* implementation dependent     */
    } pet_r1;
    struct Pchoice_pet_r2 {
                Pint            num_prompts;    /* number of prompts            */
                Ppr_switch      *prompts;       /* array of prompts             */
                int impl_dep;                   /* implementation dependent     */
    } pet_r2;
    struct Pchoice_pet_r3 {
                Pint            num_strings;    /* number of choice strings     */
                char            **strings;      /* array of choice strings      */
                int impl_dep;                   /* implementation dependent     */
    } pet_r3;
    struct Pchoice_pet_r4 {
                Pint            num_strings;    /* number of choice strings     */
                char            **strings;      /* array of choice strings      */
                int impl_dep;                   /* implementation dependent     */
    } pet_r4;
    struct Pchoice_pet_r5 {
                Pint            struct_id;      /* struct identifier            */
                Pint            num_pick_ids;   /* number of pick identifiers   */
                Pint            *pick_ids;      /* array of pick identifiers    */
                int impl_dep;                   /* implementation dependent     */
    } pet_r5;
    int impl_dep;                               /* implementation defined PET's */
  } pets;
} Pchoice_data;


typedef struct {
  union Ppick_pets {
    struct Ppick_pet_r1 {
                int impl_dep;    /* implementation dependent     */
    } pet_r1;
    int impl_dep;                /* implementation defined PET's */
  } pets;
} Ppick_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint  in_buf_size;                    /* input buffer size            */
  Pint  init_pos;                       /* initial editing position     */
  union Pstring_pets {
        struct Pstring_pet_r1 {
                    int impl_dep;       /* implementation dependent     */
        } pet_r1;
        int impl_dep;                   /* implementation defined PET's */
  } pets;
} Pstring_data;


typedef struct {
  union Ploc3_pets {
    struct Ploc3_pet_r1 {
                int impl_dep;                                   /* impl. dependent      */
    } pet_r1;
    struct Ploc3_pet_r2 {
                int impl_dep;                                   /* impl. dependent      */
    } pet_r2;
    struct Ploc3_pet_r3 {
                int impl_dep;                                   /* impl. dependent      */
    } pet_r3;
    struct Ploc3_pet_r4 {
                Pline_attrs          line_attrs;                /* polyline attributes  */
                int impl_dep;                                   /* impl. dependent      */
    } pet_r4;
    struct Ploc3_pet_r5 {
                Pline_fill_ctrl_flag line_fill_ctrl_flag;       /* control flag         */
                union Ploc3_attrs {
                                     Pline_attrs          line_attrs;
                                                          /* polyline attributes  */
                                     Pint_attrs           int_attrs;
                                                          /* interior attributes  */
                                     struct Ploc3_fill_set {
                                                          Pint_attrs  int_attrs;
                                                          /* interior attributes  */
                                                          Pedge_attrs  edge_attrs;
                                                          /* edge attributes      */
                                     } fill_set;
                } attrs;
                int impl_dep;                                   /* impl. dependent      */
    } pet_r5;
    struct Ploc3_pet_r6 {
                int impl_dep;                                   /* impl. dependent      */
    } pet_r6;
    int impl_dep;                                               /* impl. defined PET's  */
  } pets;
} Ploc_data3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint    in_buf_size;                              /* input buffer size           */
  Pint    init_pos;                                 /* initial editing position    */
  Pfloat  x_interval;                               /* x interval                  */
  Pfloat  y_interval;                               /* y interval                  */
  Pfloat  z_interval;                               /* z interval                  */
  Pfloat  time_interval;                            /* time interval               */
  union Pstroke3_pets {
        struct Pstroke3_pet_r1 {
                    int impl_dep;                   /* implementation dependent    */
        } pet_r1;
        struct Pstroke3_pet_r2 {
                    int impl_dep;                   /* implementation dependent    */
        } pet_r2;
        struct Pstroke3_pet_r3 {
                    Pmarker_attrs  marker_attrs;    /* marker attributes           */
                    int impl_dep;                   /* implementation dependent    */
        } pet_r3;
        struct Pstroke3_pet_r4 {
                    Pline_attrs    line_attrs;      /* line attributes             */
                    int impl_dep;                   /* implementation dependent    */
        } pet_r4;
        int impl_dep;                               /* implementation defined PET's */
  } pets;
} Pstroke_data3;

typedef struct {
  Pfloat  low_value;                    /* low value of valuator range  */
  Pfloat  high_value;                   /* high value of valuator range */
  union Pval3_pets {
        struct Pval3_pet_r1 {
                    int impl_dep;       /* implementation dependent     */
        } pet_r1;
        int impl_dep;                   /* implementation defined PET's */
  } pets;
} Pval_data3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  union Pchoice3_pets {
    struct Pchoice3_pet_r1 {
                int impl_dep;                   /* implementation dependent     */
    } pet_r1;
    struct Pchoice3_pet_r2 {
                Pint            num_prompts;    /* number of prompts            */
                Ppr_switch      *prompts;       /* array of prompts             */
                int impl_dep;                   /* implementation dependent     */
    } pet_r2;
    struct Pchoice3_pet_r3 {
                Pint            num_strings;    /* number of choice strings     */
                char            **strings;      /* array of choice strings      */
                int impl_dep;                   /* implementation dependent     */
    } pet_r3;
    struct Pchoice3_pet_r4 {
                Pint            num_strings;    /* number of choice strings     */
                char            **strings;      /* array of choice strings      */
                int impl_dep;                   /* implementation dependent     */
    } pet_r4;
    struct Pchoice3_pet_r5 {
                Pint            struct_id;      /* struct identifier            */
                Pint            num_pick_ids;   /* number of pick identifiers   */
                Pint            *pick_ids;      /* array of pick identifiers    */
                int impl_dep;                   /* implementation dependent     */
    } pet_r5;
    int impl_dep;                               /* implementation defined PET's */
  } pets;
} Pchoice_data3;


typedef struct {
  union Ppick3_pets {
    struct Ppick3_pet_r1 {
                int impl_dep;    /* implementation dependent     */
    } pet_r1;
    int impl_dep;                /* implementation defined PET's */
  } pets;
} Ppick_data3;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef struct {
  Pint  in_buf_size;                    /* input buffer size            */
  Pint  init_pos;                       /* initial editing position     */
  union Pstring3_pets {
        struct Pstring3_pet_r1 {
                    int impl_dep;       /* implementation dependent     */
        } pet_r1;
        int impl_dep;                   /* implementation defined PET's */
  } pets;
} Pstring_data3;


typedef union {
  struct Pgdp_r1 {
              int impl_dep;     /* registration dependent      */
  } gdp_r1;
  Pdata       unsupp;           /* unsupported GDP data record */
  int impl_dep;                 /* implementation defined      */
} Pgdp_data;


typedef union {
  struct Pgdp3_r1 {
              int impl_dep;     /* registration dependent       */
  } gdp3_r1;
  Pdata       unsupp;           /* unsupported GDP3 data record */
  int impl_dep;                 /* implementation defined       */
} Pgdp_data3;


typedef union {
  struct Pgse_r1 {
              int impl_dep;     /* registration dependent      */
  } gse_r1;
  Pdata       unsupp;           /* unsupported GSE data record */
  int impl_dep;                 /* implementation defined      */
} Pgse_data;


typedef union {
  struct Pescape_out_r1 {
              int impl_dep;     /* registration dependent */
  } escape_out_r1;
  int impl_dep;                 /* implementation defined */
} Pescape_out_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
typedef union {
  struct Pescape_in_r1 {
                int impl_dep;      /* registration dependent */
  } escape_in_r1;
  int impl_dep;                    /* implementation defined */
} Pescape_in_data;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

```
typedef union {
  Pint              int_data;                    /* integer valued data     */
  Pfloat            float_data;                  /* float valued data       */
  Ppoint_list3      point_list3;                 /* list of 3d points       */
  Ppoint_list       point_list;                  /* list of 2d points       */
  Ppoint_list_list3 point_list_list3;            /* list of 3d point lists */
  Ppoint_list_list  point_list_list;             /* list of 2d point lists */
  struct Pelem_text3 {
                    Ppoint3         pos;         /* text position           */
                    Pvec3           dir[2];      /* direction vectors       */
                    char            *char_string; /* character string       */
  } text3;
  struct Pelem_text {
                    Ppoint          pos;         /* text position           */
                    char            *char_string; /* character string       */
  } text;
  struct Pelem_anno_text_rel3 {
                    Ppoint3         ref_point;   /* reference point         */
                    Pvec3           offset;      /* anno. offset            */
                    char            *char_string; /* character string       */
  } anno_text_rel3;
  struct Pelem_anno_text_rel {
                    Ppoint          ref_point;   /* reference point         */
                    Pvec            offset;      /* anno. offset            */
                    char            *char_string; /* character string       */
  } anno_text_rel;
  struct Pelem_cell_array3 {
                    Pparal          paral;       /* parallelogram           */
                    Ppat_rep        colr_array;  /* colour array            */
  } cell_array3;
  struct Pelem_cell_array {
                    Prect           rect;        /* rectangle               */
                    Ppat_rep        colr_array;  /* colour array            */
  } cell_array;
  struct Pelem_gdp3 {
                    Pint            id;          /* GDP3 id                 */
                    Ppoint_list3    point_list;  /* point list              */
                    Pgdp_data3      data;        /* data record             */
  } gdp3;
  struct Pelem_gdp {
                    Pint            id;          /* GDP id                  */
                    Ppoint_list     point_list;  /* point list              */
                    Pgdp_data       data;        /* data record             */
  } gdp;
  Ptext_prec        text_prec;                   /* text precision          */
  Pvec              char_up_vec;                 /* char up vector          */
  Ptext_path        text_path;                   /* text path               */
  Ptext_align       text_align;                  /* text alignment          */
  Pint_style        int_style;                   /* interior style          */
  Pedge_flag        edge_flag;                   /* edge flag               */
  Ppoint            pat_ref_point;               /* pattern ref. point      */
  Pfloat_size       pat_size;                    /* pattern size            */
  struct Pelem_pat_ref_point_vecs {
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
                    Ppoint3             ref_point;      /* pattern ref. point     */
                    Pvec3               ref_vec[2];     /* vectors                */
  } pat_ref_point_vecs;
  Pint_list         names;                              /* name sets              */
  struct Pelem_asf {
                    Paspect             id;             /* attribute id           */
                    Pasf                source;         /* attribute source       */
  } asf;
  struct Pelem_local_tran3 {
                    Pcompose_type       compose_type;   /* composition type       */
                    Pmatrix3            matrix;         /* tran. matrix           */
  } local_tran3;
  struct Pelem_local_tran {
                    Pcompose_type       compose_type;   /* composition type       */
                    Pmatrix             matrix;         /* tran. matrix           */
  } local_tran;
  Pmatrix3          global_tran3;                       /* global transform3      */
  Pmatrix           global_tran;                        /* global transform       */
  struct Pelem_model_clip3 {
                    Pint                op;             /* operator               */
                    Phalf_space_list3   half_spaces;    /* half space list        */
  } model_clip3;
  struct Pelem_model_clip {
                    Pint                op;             /* operator               */
                    Phalf_space_list    half_spaces;    /* half space list        */
  } model_clip;
  Pclip_ind         clip_ind;                           /* clipping indicator     */
  Pdata             appl_data;                          /* application data       */
  struct Pelem_gse {
                    Pint                id;             /* GSE id                 */
                    Pgse_data           data;           /* GSE data record        */
  } gse;
} Pelem_data;
```

A.3 External functions

```
/* OPEN PHIGS */
extern void popen_phigs(
    const char    *err_file,   /* name of error file                                  */
    size_t        mem_units    /* size_t units of memory available for buffer space   */
);

/* CLOSE PHIGS */
extern void pclose_phigs(
      void
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* OPEN WORKSTATION */
extern void popen_ws(
    Pint        ws_id,      /* workstation identifier   */
    const void  *conn_id,   /* connection identifier    */
    Pint        ws_type     /* workstation type         */
);

/* CLOSE WORKSTATION */
extern void pclose_ws(
    Pint  ws_id   /* workstation identifier   */
);

/* REDRAW ALL STRUCTURES */
extern void predraw_all_structs(
    Pint        ws_id,      /* workstation identifier   */
    Pctrl_flag  ctrl_flag   /* control flag             */
);

/* UPDATE WORKSTATION */
extern void pupd_ws(
    Pint         ws_id,       /* workstation identifier     */
    Pregen_flag  regen_flag   /* update regeneration flag   */
);

/* SET DISPLAY UPDATE STATE */
extern void pset_disp_upd_st(
    Pint         ws_id,      /* workstation identifier   */
    Pdefer_mode  def_mode,   /* deferral mode            */
    Pmod_mode    mod_mode    /* modification mode        */
);

/* MESSAGE */
extern void pmessage(
    Pint         ws_id,     /* workstation identifier   */
    const char   *message   /* message string           */
);

/* POLYLINE 3 */
extern void ppolyline3(
    const Ppoint_list3   *point_list   /* list of points   */
);

/* POLYLINE */
extern void ppolyline(
    const Ppoint_list   *point_list   /* list of points   */
);

/* POLYMARKER 3 */
extern void ppolymarker3(
    const Ppoint_list3   *point_list   /* list of points   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* POLYMARKER */
extern void ppolymarker(
    const Ppoint_list   *point_list   /* list of points   */
);

/* TEXT 3 */
extern void ptext3(
    const Ppoint3   *text_pos,     /* text position           */
    const Pvec3     text_dir[2],   /* text direction vectors  */
    const char      *char_string   /* character string        */
);

/* TEXT */
extern void ptext(
    const Ppoint   *text_pos,     /* text position      */
    const char     *char_string   /* character string   */
);

/* ANNOTATION TEXT RELATIVE 3 */
extern void panno_text_rel3(
    const Ppoint3   *ref_point,    /* reference point              */
    const Pvec3     *offset,       /* annotation offset            */
    const char      *char_string   /* annotation character string  */
);

/* ANNOTATION TEXT RELATIVE */
extern void panno_text_rel(
    const Ppoint   *ref_point,    /* reference point              */
    const Pvec     *offset,       /* annotation offset            */
    const char     *char_string   /* annotation character string  */
);

/* FILL AREA 3 */
extern void pfill_area3(
    const Ppoint_list3   *point_list   /* list of points   */
);

/* FILL AREA */
extern void pfill_area(
    const Ppoint_list   *point_list   /* list of points   */
);

/* FILL AREA SET 3 */
extern void pfill_area_set3(
    const Ppoint_list_list3   *point_list_list   /* list of point lists   */
);

/* FILL AREA SET */
extern void pfill_area_set(
    const Ppoint_list_list   *point_list_list   /* list of point lists   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* CELL ARRAY 3 */
extern void pcell_array3(
    const Pparal     *paral,       /* cell parallelogram  */
    const Ppat_rep   *colr_array   /* colour array        */
);

/* CELL ARRAY */
extern void pcell_array(
    const Prect      *rect,        /* cell rectangle   */
    const Ppat_rep   *colr_array   /* colour array     */
);

/* GENERALIZED DRAWING PRIMITIVE 3 */
extern void pgdp3(
    const Ppoint_list3   *point_list,   /* list of points           */
    Pint                 gdp3_id,       /* gdp function identifier  */
    const Pgdp_data3     *gdp_data      /* gdp data record          */
);

/* GENERALIZED DRAWING PRIMITIVE */
extern void pgdp(
    const Ppoint_list   *point_list,   /* list of points           */
    Pint                gdp_id,        /* gdp function identifier  */
    const Pgdp_data     *gdp_data      /* gdp data record          */
);

/* SET POLYLINE INDEX */
extern void pset_line_ind(
    Pint   line_ind   /* polyline index   */
);

/* SET POLYMARKER INDEX */
extern void pset_marker_ind(
    Pint   marker_ind   /* polymarker index   */
);

/* SET TEXT INDEX */
extern void pset_text_ind(
    Pint   text_ind   /* text index   */
);

/* SET INTERIOR INDEX */
extern void pset_int_ind(
    Pint   int_ind   /* interior index   */
);

/* SET EDGE INDEX */
extern void pset_edge_ind(
    Pint   edge_ind   /* edge index   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET LINETYPE */
extern void pset_linetype(
    Pint   linetype   /* linetype   */
);

/* SET LINEWIDTH SCALE FACTOR */
extern void pset_linewidth(
    Pfloat   linewidth   /* linewidth scale factor   */
);

/* SET POLYLINE COLOUR INDEX */
extern void pset_line_colr_ind(
    Pint   line_colr_ind   /* polyline colour index   */
);

/* SET MARKER TYPE */
extern void pset_marker_type(
    Pint   marker_type   /* marker type   */
);

/* SET MARKER SIZE SCALE FACTOR */
extern void pset_marker_size(
    Pfloat   marker_size   /* marker size scale factor   */
);

/* SET POLYMARKER COLOUR INDEX */
extern void pset_marker_colr_ind(
    Pint   marker_colr_ind   /* polymarker colour index   */
);

/* SET TEXT FONT */
extern void pset_text_font(
    Pint   font   /* text font   */
);

/* SET TEXT PRECISION */
extern void pset_text_prec(
    Ptext_prec   prec   /* text precision   */
);

/* SET CHARACTER EXPANSION FACTOR */
extern void pset_char_expan(
    Pfloat   char_expan   /* character expansion factor   */
);

/* SET CHARACTER SPACING */
extern void pset_char_space(
    Pfloat   char_space   /* character spacing   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577535 450 ■


```
/* SET TEXT COLOUR INDEX */
extern void pset_text_colr_ind(
    Pint   text_colr_ind   /* text colour index   */
);

/* SET CHARACTER HEIGHT */
extern void pset_char_ht(
    Pfloat   char_ht   /* character height   */
);

/* SET CHARACTER UP VECTOR */
extern void pset_char_up_vec(
    const Pvec   *char_up_vec   /* character up vector   */
);

/* SET TEXT PATH */
extern void pset_text_path(
    Ptext_path   text_path   /* text path   */
);

/* SET TEXT ALIGNMENT */
extern void pset_text_align(
    const Ptext_align   *text_align   /* text alignment   */
);

/* SET ANNOTATION TEXT CHARACTER HEIGHT */
extern void pset_anno_char_ht(
    Pfloat   char_ht   /* character height   */
);

/* SET ANNOTATION TEXT CHARACTER UP VECTOR */
extern void pset_anno_char_up_vec(
    const Pvec   *char_up_vec   /* character up vector   */
);

/* SET ANNOTATION TEXT PATH */
extern void pset_anno_path(
    Ptext_path   text_path   /* text path   */
);

/* SET ANNOTATION TEXT ALIGNMENT */
extern void pset_anno_align(
    const Ptext_align   *text_align   /* text alignment   */
);

/* SET ANNOTATION STYLE */
extern void pset_anno_style(
    Pint   anno_style   /* annotation style   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SET INTERIOR STYLE */
extern void pset_int_style(
    Pint_style   int_style   /* interior style   */
);

/* SET INTERIOR STYLE INDEX */
extern void pset_int_style_ind(
    Pint   int_style_ind   /* interior style index   */
);

/* SET INTERIOR COLOUR INDEX */
extern void pset_int_colr_ind(
    Pint   int_colr_ind   /* interior colour index   */
);

/* SET EDGE FLAG */
extern void pset_edge_flag(
    Pedge_flag   edge_flag   /* edge flag   */
);

/* SET EDGETYPE */
extern void pset_edgetype(
    Pint   edgetype   /* edgetype   */
);

/* SET EDGEWIDTH SCALE FACTOR */
extern void pset_edgewidth(
    Pfloat   edgewidth   /* edgewidth scale factor   */
);

/* SET EDGE COLOUR INDEX */
extern void pset_edge_colr_ind(
    Pint   edge_colr_ind   /* edge colour index   */
);

/* SET PATTERN SIZE */
extern void pset_pat_size(
    const Pfloat_size   *pat_size   /* pattern size   */
);

/* SET PATTERN REFERENCE POINT AND VECTORS */
extern void pset_pat_ref_point_vecs(
    const Ppoint3   *pat_ref_point,   /* pattern reference point                    */
    const Pvec3     pat_ref_vec[2]    /* reference vectors: 0 = X axis; 1 = Y axis  */
);

/* SET PATTERN REFERENCE POINT */
extern void pset_pat_ref_point(
    const Ppoint   *pat_ref_point   /* pattern reference point   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* ADD NAMES TO SET */
extern void padd_names_set(
    const Pint_list   *names   /* name set to be added   */
);

/* REMOVE NAMES FROM SET */
extern void premove_names_set(
    const Pint_list   *names   /* name set to be removed   */
);

/* SET INDIVIDUAL ASF */
extern void pset_indiv_asf(
    Paspect   asf_id,      /* attribute source identifier   */
    Pasf      asf_source   /* attribute source              */
);

/* SET POLYLINE REPRESENTATION */
extern void pset_line_rep(
    Pint                ws_id,         /* workstation identifier    */
    Pint                line_ind,      /* polyline bundle index     */
    const Pline_bundle  *line_bundle   /* polyline representation   */
);

/* SET POLYMARKER REPRESENTATION */
extern void pset_marker_rep(
    Pint                   ws_id,          /* workstation identifier      */
    Pint                   marker_ind,     /* polymarker bundle index     */
    const Pmarker_bundle   *marker_bundle  /* polymarker representation    */
);

/* SET TEXT REPRESENTATION */
extern void pset_text_rep(
    Pint                ws_id,         /* workstation identifier   */
    Pint                text_ind,      /* text bundle index        */
    const Ptext_bundle  *text_bundle   /* text representation      */
);

/* SET INTERIOR REPRESENTATION */
extern void pset_int_rep(
    Pint                ws_id,        /* workstation identifier    */
    Pint                int_ind,      /* interior bundle index     */
    const Pint_bundle   *int_bundle   /* interior representation   */
);

/* SET EDGE REPRESENTATION */
extern void pset_edge_rep(
    Pint                ws_id,         /* workstation identifier   */
    Pint                edge_ind,      /* edge bundle index        */
    const Pedge_bundle  *edge_bundle   /* edge representation      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET PATTERN REPRESENTATION */
extern void pset_pat_rep(
    Pint            ws_id,        /* workstation identifier   */
    Pint            pat_ind,      /* pattern bundle index     */
    const Ppat_rep  *pat_bundle   /* pattern representation   */
);

/* SET COLOUR REPRESENTATION */
extern void pset_colr_rep(
    Pint             ws_id,      /* workstation identifier   */
    Pint             colr_ind,   /* colour bundle index      */
    const Pcolr_rep  *colr_rep   /* colour representation    */
);

/* SET HIGHLIGHTING FILTER */
extern void pset_highl_filter(
    Pint           ws_id,    /* workstation identifier   */
    const Pfilter  *filter   /* highlighting filter      */
);

/* SET INVISIBILITY FILTER */
extern void pset_invis_filter(
    Pint           ws_id,    /* workstation identifier   */
    const Pfilter  *filter   /* invisibility filter      */
);

/* SET COLOUR MODEL */
extern void pset_colr_model(
    Pint   ws_id,       /* workstation identifier   */
    Pint   colr_model   /* colour model             */
);

/* SET HLHSR IDENTIFIER */
extern void pset_hlhsr_id(
    Pint   hlhsr_id   /* HLHSR identifier   */
);

/* SET HLHSR MODE */
extern void pset_hlhsr_mode(
    Pint   ws_id,       /* workstation identifier   */
    Pint   hlhsr_mode   /* HLHSR mode               */
);

/* SET LOCAL TRANSFORMATION 3 */
extern void pset_local_tran3(
    Pmatrix3       local_tran,    /* local transformation matrix   */
    Pcompose_type  compose_type   /* composition type              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET LOCAL TRANSFORMATION */
extern void pset_local_tran(
    Pmatrix         local_tran,    /* local transformation matrix   */
    Pcompose_type   compose_type   /* composition type              */
);

/* SET GLOBAL TRANSFORMATION 3 */
extern void pset_global_tran3(
    Pmatrix3   global_tran   /* global transformation matrix   */
);

/* SET GLOBAL TRANSFORMATION */
extern void pset_global_tran(
    Pmatrix   global_tran   /* global transformation matrix   */
);

/* SET MODELLING CLIPPING VOLUME 3 */
extern void pset_model_clip_vol3(
    Pint                      op,           /* operator             */
    const Phalf_space_list3   *half_spaces  /* list of half spaces  */
);

/* SET MODELLING CLIPPING VOLUME */
extern void pset_model_clip_vol(
    Pint                    op,           /* operator             */
    const Phalf_space_list  *half_spaces  /* list of half spaces  */
);

/* SET MODELLING CLIPPING INDICATOR */
extern void pset_model_clip_ind(
    Pclip_ind   clip_ind   /* clipping indicator   */
);

/* RESTORE MODELLING CLIPPING VOLUME */
extern void prestore_model_clip_vol(
          void
);

/* SET VIEW INDEX */
extern void pset_view_ind(
    Pint   view_ind   /* view index   */
);

/* SET VIEW REPRESENTATION 3 */
extern void pset_view_rep3(
    Pint               ws_id,      /* workstation identifier   */
    Pint               view_ind,   /* view index               */
    const Pview_rep3   *view_rep   /* view representation      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET VIEW REPRESENTATION */
extern void pset_view_rep(
    Pint             ws_id,      /* workstation identifier   */
    Pint             view_ind,   /* view index               */
    const Pview_rep  *view_rep   /* view representation      */
);

/* SET VIEW TRANSFORMATION INPUT PRIORITY */
extern void pset_view_tran_in_pri(
    Pint       ws_id,          /* workstation identifier   */
    Pint       view_ind,       /* view index               */
    Pint       ref_view_ind,   /* reference view index     */
    Prel_pri   rel_pri         /* relative priority        */
);

/* SET WORKSTATION WINDOW 3 */
extern void pset_ws_win3(
    Pint           ws_id,           /* workstation identifier      */
    const Plimit3  *ws_win_limits   /* workstation window limits   */
);

/* SET WORKSTATION WINDOW */
extern void pset_ws_win(
    Pint           ws_id,           /* workstation identifier      */
    const Plimit   *ws_win_limits   /* workstation window limits   */
);

/* SET WORKSTATION VIEWPORT 3 */
extern void pset_ws_vp3(
    Pint           ws_id,           /* workstation identifier       */
    const Plimit3  *ws_vp_limits    /* workstation viewport limits  */
);

/* SET WORKSTATION VIEWPORT */
extern void pset_ws_vp(
    Pint           ws_id,           /* workstation identifier       */
    const Plimit   *ws_vp_limits    /* workstation viewport limits  */
);

/* TRANSLATE 3 */
extern void ptranslate3(
    const Pvec3    *trans_vec,   /* translation vector          */
    Pint           *err_ind,     /* OUT error indicator         */
    Pmatrix3       result_tran   /* OUT transformation matrix   */
);

/* TRANSLATE */
extern void ptranslate(
    const Pvec     *trans_vec,   /* translation vector          */
    Pint           *err_ind,     /* OUT error indicator         */
    Pmatrix        result_tran   /* OUT transformation matrix   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* SCALE 3 */
extern void pscale3(
    const Pvec3   *scale_vec,   /* scale factor vector          */
    Pint          *err_ind,     /* OUT error indicator          */
    Pmatrix3      result_tran   /* OUT transformation matrix    */
);

/* SCALE */
extern void pscale(
    const Pvec   *scale_vec,   /* scale factor vector          */
    Pint         *err_ind,     /* OUT error indicator          */
    Pmatrix      result_tran   /* OUT transformation matrix    */
);

/* ROTATE X */
extern void protate_x(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);

/* ROTATE Y */
extern void protate_y(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);

/* ROTATE Z */
extern void protate_z(
    Pfloat     angle,        /* rotation angle               */
    Pint       *err_ind,     /* OUT error indicator          */
    Pmatrix3   result_tran   /* OUT transformation matrix    */
);

/* ROTATE */
extern void protate(
    Pfloat    angle,        /* rotation angle               */
    Pint      *err_ind,     /* OUT error indicator          */
    Pmatrix   result_tran   /* OUT transformation matrix    */
);

/* COMPOSE MATRIX 3 */
extern void pcompose_matrix3(
    Pmatrix3   tran_a,       /* transformation matrix a     */
    Pmatrix3   tran_b,       /* transformation matrix b     */
    Pint       *err_ind,     /* OUT error indicator         */
    Pmatrix3   result_tran   /* OUT result matrix           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* COMPOSE MATRIX */
extern void pcompose_matrix(
    Pmatrix   tran_a,       /* transformation matrix a   */
    Pmatrix   tran_b,       /* transformation matrix b   */
    Pint      *err_ind,     /* OUT error indicator       */
    Pmatrix   result_tran   /* OUT result matrix         */
);

/* TRANSFORM POINT 3 */
extern void ptran_point3(
    const Ppoint3   *point,     /* point                    */
    Pmatrix3        tran,       /* transformation matrix    */
    Pint            *err_ind,   /* OUT error indicator      */
    Ppoint3         *result     /* OUT transformed point    */
);

/* TRANSFORM POINT */
extern void ptran_point(
    const Ppoint   *point,     /* point                     */
    Pmatrix        tran,       /* transformation matrix     */
    Pint           *err_ind,   /* OUT error indicator       */
    Ppoint         *result     /* OUT transformed point     */
);

/* BUILD TRANSFORMATION MATRIX 3 */
extern void pbuild_tran_matrix3(
    const Ppoint3   *point,       /* fixed point               */
    const Pvec3     *shift_vec,   /* shift vector              */
    Pfloat          x_angle,      /* rotation angle X          */
    Pfloat          y_angle,      /* rotation angle Y          */
    Pfloat          z_angle,      /* rotation angle Z          */
    const Pvec3     *scale_vec,   /* scale vector              */
    Pint            *err_ind,     /* OUT error indicator       */
    Pmatrix3        result_tran   /* OUT transformation matrix */
);

/* BUILD TRANSFORMATION MATRIX */
extern void pbuild_tran_matrix(
    const Ppoint   *point,       /* fixed point               */
    const Pvec     *shift_vec,   /* shift vector              */
    Pfloat         angle,        /* rotation angle            */
    const Pvec     *scale_vec,   /* scale vector              */
    Pint           *err_ind,     /* OUT error indicator       */
    Pmatrix        result_tran   /* OUT transformation matrix */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* COMPOSE TRANSFORMATION MATRIX 3 */
extern void pcompose_tran_matrix3(
    Pmatrix3       tran,         /* transformation matrix      */
    const Ppoint3  *point,       /* fixed point                */
    const Pvec3    *shift_vec,   /* shift vector               */
    Pfloat         x_angle,      /* rotation angle X           */
    Pfloat         y_angle,      /* rotation angle Y           */
    Pfloat         z_angle,      /* rotation angle Z           */
    const Pvec3    *scale_vec,   /* scale vector               */
    Pint           *err_ind,     /* OUT error indicator        */
    Pmatrix3       result_tran   /* OUT transformation matrix  */
);

/* COMPOSE TRANSFORMATION MATRIX */
extern void pcompose_tran_matrix(
    Pmatrix        tran,         /* transformation matrix      */
    const Ppoint   *point,       /* fixed point                */
    const Pvec     *shift_vec,   /* shift vector               */
    Pfloat         angle,        /* rotation angle             */
    const Pvec     *scale_vec,   /* scale vector               */
    Pint           *err_ind,     /* OUT error indicator        */
    Pmatrix        result_tran   /* OUT transformation matrix  */
);

/* EVALUATE VIEW ORIENTATION MATRIX 3 */
extern void peval_view_ori_matrix3(
    const Ppoint3  *view_ref_point,   /* view reference point          */
    const Pvec3    *view_norm_vec,    /* view plane normal             */
    const Pvec3    *view_up_vec,      /* view up vector                */
    Pint           *err_ind,          /* OUT error indicator           */
    Pmatrix3       result_tran        /* OUT view orientation matrix   */
);

/* EVALUATE VIEW ORIENTATION MATRIX */
extern void peval_view_ori_matrix(
    const Ppoint   *view_ref_point,   /* view reference point          */
    const Pvec     *view_up_vec,      /* view up vector                */
    Pint           *err_ind,          /* OUT error indicator           */
    Pmatrix        result_tran        /* OUT view orientation matrix   */
);

/* EVALUATE VIEW MAPPING MATRIX 3 */
extern void peval_view_map_matrix3(
    const Pview_map3  *mapping,     /* view mapping               */
    Pint              *err_ind,     /* OUT error indicator        */
    Pmatrix3          result_tran   /* OUT view mapping matrix    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* EVALUATE VIEW MAPPING MATRIX */
extern void peval_view_map_matrix(
    const Pview_map   *mapping,     /* view mapping              */
    Pint              *err_ind,     /* OUT error indicator       */
    Pmatrix           result_tran   /* OUT view mapping matrix   */
);

/* OPEN STRUCTURE */
extern void popen_struct(
    Pint   struct_id   /* structure identifier   */
);

/* CLOSE STRUCTURE */
extern void pclose_struct(
      void
);

/* EXECUTE STRUCTURE */
extern void pexec_struct(
    Pint   struct_id   /* structure identifier   */
);

/* LABEL */
extern void plabel(
    Pint   label_id   /* label identifier   */
);

/* APPLICATION DATA */
extern void pappl_data(
    const Pdata   *data   /* application data   */
);

/* GENERALIZED STRUCTURE ELEMENT */
extern void pgse(
    Pint              id,         /* gse identifier    */
    const Pgse_data   *gse_data   /* gse data record   */
);

/* SET EDIT MODE */
extern void pset_edit_mode(
    Pedit_mode   edit_mode   /* edit mode   */
);

/* COPY ALL ELEMENTS FROM STRUCTURE */
extern void pcopy_all_elems_struct(
    Pint   struct_id   /* structure identifier   */
);

/* SET ELEMENT POINTER */
extern void pset_elem_ptr(
    Pint   elem_ptr_value   /* element pointer value   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577545 3TT ■


```
/* OFFSET ELEMENT POINTER */
extern void poffset_elem_ptr(
    Pint   elem_ptr_offset   /* element pointer offset   */
);

/* SET ELEMENT POINTER AT LABEL */
extern void pset_elem_ptr_label(
    Pint   label_id   /* label identifier   */
);

/* DELETE ELEMENT */
extern void pdel_elem(
     void
);

/* DELETE ELEMENT RANGE */
extern void pdel_elem_range(
    Pint   elem_ptr1_value,   /* element pointer 1 value   */
    Pint   elem_ptr2_value    /* element pointer 2 value   */
);

/* DELETE ELEMENTS BETWEEN LABELS */
extern void pdel_elems_labels(
    Pint   label1_id,   /* label 1 identifier   */
    Pint   label2_id    /* label 2 identifier   */
);

/* EMPTY STRUCTURE */
extern void pempty_struct(
    Pint   struct_id   /* structure id   */
);

/* DELETE STRUCTURE */
extern void pdel_struct(
    Pint   struct_id   /* structure identifier   */
);

/* DELETE STRUCTURE NETWORK */
extern void pdel_struct_net(
    Pint        struct_id,   /* structure identifier      */
    Pref_flag   ref_flag     /* reference handling flag   */
);

/* DELETE ALL STRUCTURES */
extern void pdel_all_structs(
       void
);

/* CHANGE STRUCTURE IDENTIFIER */
extern void pchange_struct_id(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* CHANGE STRUCTURE REFERENCES */
extern void pchange_struct_refs(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);

/* CHANGE STRUCTURE IDENTIFIER AND REFERENCES */
extern void pchange_struct_id_refs(
    Pint   orig_struct_id,    /* original structure id   */
    Pint   result_struct_id   /* result structure id     */
);

/* POST STRUCTURE */
extern void ppost_struct(
    Pint     ws_id,       /* workstation identifier   */
    Pint     struct_id,   /* structure identifier     */
    Pfloat   pri          /* priority                 */
);

/* UNPOST STRUCTURE */
extern void punpost_struct(
    Pint   ws_id,      /* workstation identifier   */
    Pint   struct_id   /* structure identifier     */
);

/* UNPOST ALL STRUCTURES */
extern void punpost_all_structs(
    Pint   ws_id   /* workstation identifier   */
);

/* OPEN ARCHIVE FILE */
extern void popen_ar_file(
    Pint          archive_id,     /* archive identifier   */
    const char    *archive_file   /* archive file name    */
);

/* CLOSE ARCHIVE FILE */
extern void pclose_ar_file(
    Pint   archive_id   /* archive identifier   */
);

/* ARCHIVE STRUCTURES */
extern void par_structs(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);

/* ARCHIVE STRUCTURE NETWORKS */
extern void par_struct_nets(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* ARCHIVE ALL STRUCTURES */
extern void par_all_structs(
    Pint   archive_id   /* archive identifier   */
);

/* SET CONFLICT RESOLUTION */
extern void pset_conf_res(
    Pconf_res   archival_res,   /* archival conflict resolution    */
    Pconf_res   retrieval_res   /* retrieval conflict resolution   */
);

/* RETRIEVE STRUCTURE IDENTIFIERS */
extern void pret_struct_ids(
    Pint        archive_id,           /* archive identifier          */
    Pint        num_elems_appl_list,  /* # elems in appl list        */
    Pint        start_ind,            /* starting index              */
    Pint_list   *ids,                 /* OUT list of structure ids   */
    Pint        *num_elems_impl_list  /* OUT # elems in impl. list   */
);

/* RETRIEVE STRUCTURES */
extern void pret_structs(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);

/* RETRIEVE STRUCTURE NETWORKS */
extern void pret_struct_nets(
    Pint              archive_id,   /* archive identifier             */
    const Pint_list   *struct_ids   /* list of structure identifiers  */
);

/* RETRIEVE ALL STRUCTURES */
extern void pret_all_structs(
    Pint   archive_id   /* archive identifier   */
);

/* RETRIEVE PATHS TO ANCESTORS */
extern void pret_paths_ances(
    Pint                ar_id,       /* archive identifier       */
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* RETRIEVE PATHS TO DESCENDANTS */
extern void pret_paths_descs(
    Pint                ar_id,       /* archive identifier       */
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);

/* DELETE STRUCTURES FROM ARCHIVE */
extern void pdel_structs_ar(
    Pint             archive_id,   /* archive identifier             */
    const Pint_list  *struct_ids   /* list of structure identifiers  */
);

/* DELETE STRUCTURE NETWORKS FROM ARCHIVE */
extern void pdel_struct_nets_ar(
    Pint             archive_id,   /* archive identifier             */
    const Pint_list  *struct_ids   /* list of structure identifiers  */
);

/* DELETE ALL STRUCTURES FROM ARCHIVE */
extern void pdel_all_structs_ar(
    Pint  archive_id  /* archive identifier  */
);

/* SET PICK IDENTIFIER */
extern void pset_pick_id(
    Pint  pick_id  /* pick identifier  */
);

/* SET PICK FILTER */
extern void pset_pick_filter(
    Pint           ws_id,     /* workstation identifier  */
    Pint           pick_num,  /* pick device number      */
    const Pfilter  *filter    /* pick filter             */
);

/* INITIALIZE LOCATOR 3 */
extern void pinit_loc3(
    Pint              ws_id,          /* workstation identifier    */
    Pint              loc_num,        /* locator device number     */
    Pint              init_view_ind,  /* initial view index        */
    const Ppoint3     *init_loc_pos,  /* initial locator position  */
    Pint              pet,            /* prompt and echo type      */
    const Plimit3     *echo_vol,      /* echo volume               */
    const Ploc_data3  *loc_data       /* data record               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INITIALIZE LOCATOR */
extern void pinit_loc(
    Pint              ws_id,          /* workstation identifier     */
    Pint              loc_num,        /* locator device number      */
    Pint              init_view_ind,  /* initial view index         */
    const Ppoint      *init_loc_pos,  /* initial locator position   */
    Pint              pet,            /* prompt and echo type       */
    const Plimit      *echo_area,     /* echo area                  */
    const Ploc_data   *loc_data       /* data record                */
);

/* INITIALIZE STROKE 3 */
extern void pinit_stroke3(
    Pint                 ws_id,          /* workstation identifier   */
    Pint                 stroke_num,     /* stroke device number     */
    Pint                 init_view_ind,  /* initial view index       */
    const Ppoint_list3   *init_stroke,   /* initial stroke           */
    Pint                 pet,            /* prompt and echo type     */
    const Plimit3        *echo_vol,      /* echo volume              */
    const Pstroke_data3  *stroke_data    /* data record              */
);

/* INITIALIZE STROKE */
extern void pinit_stroke(
    Pint                ws_id,          /* workstation identifier   */
    Pint                stroke_num,     /* stroke device number     */
    Pint                init_view_ind,  /* initial view index       */
    const Ppoint_list   *init_stroke,   /* initial stroke           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Pstroke_data  *stroke_data    /* data record              */
);

/* INITIALIZE VALUATOR 3 */
extern void pinit_val3(
    Pint              ws_id,        /* workstation identifier   */
    Pint              val_num,      /* valuator device number   */
    Pfloat            init_value,   /* initial value            */
    Pint              pet,          /* prompt and echo type     */
    const Plimit3     *echo_vol,    /* echo volume              */
    const Pval_data3  *val_data     /* data record              */
);

/* INITIALIZE VALUATOR */
extern void pinit_val(
    Pint              ws_id,        /* workstation identifier   */
    Pint              val_num,      /* valuator device number   */
    Pfloat            init_value,   /* initial value            */
    Pint              pet,          /* prompt and echo type     */
    const Plimit      *echo_area,   /* echo area                */
    const Pval_data   *val_data     /* data record              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INITIALIZE CHOICE 3 */
extern void pinit_choice3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                choice_num,     /* choice device number     */
    Pin_status          init_status,    /* initial choice status    */
    Pint                init_choice,    /* initial choice           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Pchoice_data3 *choice_data    /* data record              */
);

/* INITIALIZE CHOICE */
extern void pinit_choice(
    Pint                ws_id,          /* workstation identifier   */
    Pint                choice_num,     /* choice device number     */
    Pin_status          init_status,    /* initial choice status    */
    Pint                init_choice,    /* initial choice           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Pchoice_data  *choice_data    /* data record              */
);

/* INITIALIZE PICK 3 */
extern void pinit_pick3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                pick_num,       /* pick device number       */
    Pin_status          init_status,    /* initial pick status      */
    const Ppick_path    *init_pick,     /* initial pick path        */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Ppick_data3   *pick_data,     /* data record              */
    Ppath_order         order           /* pick path order          */
);

/* INITIALIZE PICK */
extern void pinit_pick(
    Pint                ws_id,          /* workstation identifier   */
    Pint                pick_num,       /* pick device number       */
    Pin_status          init_status,    /* initial pick status      */
    const Ppick_path    *init_pick,     /* initial pick path        */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Ppick_data    *pick_data,     /* data record              */
    Ppath_order         order           /* pick path order          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INITIALIZE STRING 3 */
extern void pinit_string3(
    Pint                ws_id,          /* workstation identifier   */
    Pint                string_num,     /* string device number     */
    const char          *init_string,   /* initial string           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit3       *echo_vol,      /* echo volume              */
    const Pstring_data3 *string_data    /* data record              */
);

/* INITIALIZE STRING */
extern void pinit_string(
    Pint                ws_id,          /* workstation identifier   */
    Pint                string_num,     /* string device number     */
    const char          *init_string,   /* initial string           */
    Pint                pet,            /* prompt and echo type     */
    const Plimit        *echo_area,     /* echo area                */
    const Pstring_data  *string_data    /* data record              */
);

/* SET LOCATOR MODE */
extern void pset_loc_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          loc_num,      /* locator device number    */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);

/* SET STROKE MODE */
extern void pset_stroke_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);

/* SET VALUATOR MODE */
extern void pset_val_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          val_num,      /* valuator device number   */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);

/* SET CHOICE MODE */
extern void pset_choice_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          choice_num,   /* choice device number     */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SET PICK MODE */
extern void pset_pick_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          pick_num,     /* pick device number       */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);

/* SET STRING MODE */
extern void pset_string_mode(
    Pint          ws_id,        /* workstation identifier   */
    Pint          string_num,   /* string device number     */
    Pop_mode      op_mode,      /* operating mode           */
    Pecho_switch  echo_switch   /* echo switch              */
);

/* REQUEST LOCATOR 3 */
extern void preq_loc3(
    Pint         ws_id,        /* workstation identifier   */
    Pint         loc_num,      /* locator device number    */
    Pin_status   *in_status,   /* OUT input status         */
    Pint         *view_ind,    /* OUT view index           */
    Ppoint3      *loc_pos      /* OUT locator position     */
);

/* REQUEST LOCATOR */
extern void preq_loc(
    Pint         ws_id,        /* workstation identifier   */
    Pint         loc_num,      /* locator device number    */
    Pin_status   *in_status,   /* OUT input status         */
    Pint         *view_ind,    /* OUT view index           */
    Ppoint       *loc_pos      /* OUT locator position     */
);

/* REQUEST STROKE 3 */
extern void preq_stroke3(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint_list3  *stroke       /* OUT stroke               */
);

/* REQUEST STROKE */
extern void preq_stroke(
    Pint          ws_id,        /* workstation identifier   */
    Pint          stroke_num,   /* stroke device number     */
    Pin_status    *in_status,   /* OUT input status         */
    Pint          *view_ind,    /* OUT view index           */
    Ppoint_list   *stroke       /* OUT stroke               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* REQUEST VALUATOR */
extern void preq_val(
    Pint         ws_id,        /* workstation identifier   */
    Pint         val_num,      /* valuator device number   */
    Pin_status   *in_status,   /* OUT input status         */
    Pfloat       *value        /* OUT valuator value       */
);

/* REQUEST CHOICE */
extern void preq_choice(
    Pint         ws_id,        /* workstation identifier   */
    Pint         choice_num,   /* choice device number     */
    Pin_status   *in_status,   /* OUT choice status        */
    Pint         *choice       /* OUT requested choice     */
);

/* REQUEST PICK */
extern void preq_pick(
    Pint         ws_id,        /* workstation identifier        */
    Pint         pick_num,     /* pick device number            */
    Pint         depth,        /* max. depth of returned path   */
    Pin_status   *in_status,   /* OUT pick status               */
    Ppick_path   *pick         /* OUT requested pick path       */
);

/* REQUEST STRING */
extern void preq_string(
    Pint         ws_id,        /* workstation identifier   */
    Pint         string_num,   /* string device number     */
    Pin_status   *in_status,   /* OUT input status         */
    char         *string       /* OUT requested string     */
);

/* SAMPLE LOCATOR 3 */
extern void psample_loc3(
    Pint      ws_id,       /* workstation identifier   */
    Pint      loc_num,     /* locator device number    */
    Pint      *view_ind,   /* OUT view index           */
    Ppoint3   *loc_pos     /* OUT locator position     */
);

/* SAMPLE LOCATOR */
extern void psample_loc(
    Pint     ws_id,       /* workstation identifier   */
    Pint     loc_num,     /* locator device number    */
    Pint     *view_ind,   /* OUT view index           */
    Ppoint   *loc_pos     /* OUT locator position     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* SAMPLE STROKE 3 */
extern void psample_stroke3(
    Pint           ws_id,        /* workstation identifier   */
    Pint           stroke_num,   /* stroke device number     */
    Pint           *view_ind,    /* OUT view index           */
    Ppoint_list3   *stroke       /* OUT stroke               */
);

/* SAMPLE STROKE */
extern void psample_stroke(
    Pint           ws_id,        /* workstation identifier   */
    Pint           stroke_num,   /* stroke device number     */
    Pint           *view_ind,    /* OUT view index           */
    Ppoint_list    *stroke       /* OUT stroke               */
);

/* SAMPLE VALUATOR */
extern void psample_val(
    Pint     ws_id,     /* workstation identifier   */
    Pint     val_num,   /* valuator device number   */
    Pfloat   *value     /* OUT valuator value       */
);

/* SAMPLE CHOICE */
extern void psample_choice(
    Pint         ws_id,              /* workstation identifier    */
    Pint         choice_num,         /* choice device number      */
    Pin_status   *choice_in_status,  /* OUT choice input status   */
    Pint         *choice             /* OUT choice                */
);

/* SAMPLE PICK */
extern void psample_pick(
    Pint         ws_id,              /* workstation identifier       */
    Pint         pick_num,           /* pick device number           */
    Pint         depth,              /* max. depth of returned path  */
    Pin_status   *pick_in_status,    /* OUT pick input status        */
    Ppick_path   *pick               /* OUT pick path                */
);

/* SAMPLE STRING */
extern void psample_string(
    Pint    ws_id,        /* workstation identifier  */
    Pint    string_num,   /* string device number    */
    char    *string       /* OUT string              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* AWAIT EVENT */
extern void pawait_event(
    Pfloat      timeout,      /* timeout (seconds)                */
    Pint        *ws_id,       /* OUT workstation identifier       */
    Pin_class   *dev_class,   /* OUT device class                 */
    Pint        *in_num       /* OUT logical input device number  */
);

/* FLUSH DEVICE EVENTS */
extern void pflush_events(
    Pint        ws_id,       /* workstation identifier       */
    Pin_class   dev_class,   /* device class                 */
    Pint        in_num       /* logical input device number  */
);

/* GET LOCATOR 3 */
extern void pget_loc3(
    Pint      *view_ind,   /* OUT view index        */
    Ppoint3   *loc_pos     /* OUT locator position  */
);

/* GET LOCATOR */
extern void pget_loc(
    Pint     *view_ind,   /* OUT view index        */
    Ppoint   *loc_pos     /* OUT locator position  */
);

/* GET STROKE 3 */
extern void pget_stroke3(
    Pint           *view_ind,   /* OUT view index   */
    Ppoint_list3   *stroke      /* OUT stroke       */
);

/* GET STROKE */
extern void pget_stroke(
    Pint          *view_ind,   /* OUT view index   */
    Ppoint_list   *stroke      /* OUT stroke       */
);

/* GET VALUATOR */
extern void pget_val(
    Pfloat   *value   /* OUT valuator value   */
);

/* GET CHOICE */
extern void pget_choice(
    Pin_status   *in_status,   /* OUT choice status   */
    Pint         *choice       /* OUT choice          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* GET PICK */
extern void pget_pick(
    Pint        depth,        /* max. depth of returned path   */
    Pin_status  *in_status,   /* OUT pick status               */
    Ppick_path  *pick         /* OUT pick path                 */
);

/* GET STRING */
extern void pget_string(
    char    *string   /* OUT string   */
);

/* WRITE ITEM TO METAFILE */
extern void pwrite_item(
    Pint              ws_id,              /* workstation identifier    */
    Pint              item_type,          /* item type                 */
    Pint              item_data_length,   /* item data record length   */
    const Pitem_data  *item_data          /* item data record          */
);

/* GET ITEM TYPE FROM METAFILE */
extern void pget_item_type(
    Pint  ws_id,              /* workstation identifier       */
    Pint  *item_type,         /* OUT item type                */
    Pint  *item_data_length   /* OUT item data record length  */
);

/* READ ITEM FROM METAFILE */
extern void pread_item(
    Pint        ws_id,                  /* workstation identifier        */
    Pint        max_item_data_length,   /* max item data record length   */
    Pitem_data  *item_data              /* OUT item data record          */
);

/* INTERPRET ITEM */
extern void pinterpret_item(
    Pint              type,               /* item type                 */
    Pint              item_data_length,   /* item data record length   */
    const Pitem_data  *item_data          /* item data record          */
);

/* INQUIRE SYSTEM STATE VALUE */
extern void pinq_sys_st(
    Psys_st   *sys_st   /* OUT the system state   */
);

/* INQUIRE WORKSTATION STATE VALUE */
extern void pinq_ws_st(
    Pws_st   *ws_st   /* OUT workstation state   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE STRUCTURE STATE VALUE */
extern void pinq_struct_st(
    Pstruct_st   *struct_st   /* OUT structure state   */
);

/* INQUIRE ARCHIVE STATE VALUE */
extern void pinq_ar_st(
    Par_st   *ar_st   /* OUT archive state   */
);

/* INQUIRE LIST OF AVAILABLE WORKSTATION TYPES */
extern void pinq_list_avail_ws_types(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *types,                /* OUT list of ws types       */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);

/* INQUIRE PHIGS FACILITIES */
extern void pinq_phigs_facs(
    Pint        num_elems_appl_list,    /* # elems in appl list                    */
    Pint        start_ind,              /* starting index                          */
    Pint        *err_ind,               /* OUT error indicator                     */
    Pint        *max_open_ws,           /* OUT max. # simult. open ws              */
    Pint        *max_open_ar,           /* OUT max. # simult. open archive files   */
    Pint        *num_avail_names,       /* OUT # available names for name sets     */
    Pint_list   *char_sets,             /* OUT list of character sets              */
    Pint        *num_elems_impl_list,   /* OUT # elems in impl list                */
    Pint        *iss_norm_max,          /* OUT ISS max. length of normal filter list   */
    Pint        *iss_inv_max            /* OUT ISS max. length of inverted filter list */
);

/* INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES */
extern void pinq_gse_facs(
    Pint              num_elems_appl_list,   /* # elems in appl list                */
    Pint              start_ind,             /* starting index                      */
    Pint              *err_ind,              /* OUT error indicator                 */
    Pgse_id_dep_list  *gse,                  /* OUT list of GSE ids and dependencies */
    Pint              *num_elems_impl_list   /* OUT # elems in impl list            */
);

/* INQUIRE MODELLING CLIPPING FACILITIES */
extern void pinq_model_clip_facs(
    Pint        num_elems_appl_list,   /* # elems in appl list          */
    Pint        start_ind,             /* starting index                */
    Pint        *err_ind,              /* OUT error indicator           */
    Pint        *num_planes,           /* OUT number of distinct planes */
    Pint_list   *ops,                  /* OUT list of operators         */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE EDIT MODE */
extern void pinq_edit_mode(
    Pint         *err_ind,    /* OUT error indicator   */
    Pedit_mode   *edit_mode   /* OUT edit mode         */
);

/* INQUIRE SET OF OPEN WORKSTATIONS */
extern void pinq_open_wss(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *open_ws_ids,          /* OUT list of ws ids         */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);

/* INQUIRE STRUCTURE IDENTIFIERS */
extern void pinq_struct_ids(
    Pint        num_elems_appl_list,   /* # elems in appl list       */
    Pint        start_ind,             /* starting index             */
    Pint        *err_ind,              /* OUT error indicator        */
    Pint_list   *struct_ids,           /* OUT list of structure ids  */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list   */
);

/* INQUIRE ARCHIVE FILES */
extern void pinq_ar_files(
    Pstore          store,       /* handle to Store object                */
    Pint            *err_ind,    /* OUT error indicator                   */
    Par_file_list   **ar_files   /* OUT list of archive file names and ids */
);

/* INQUIRE CONFLICT RESOLUTION */
extern void pinq_conf_res(
    Pint        *err_ind,       /* OUT error indicator       */
    Pconf_res   *archive_res,   /* OUT archival resolution   */
    Pconf_res   *retrieve_res   /* OUT retrieval resolution  */
);

/* INQUIRE ALL CONFLICTING STRUCTURES */
extern void pinq_all_conf_structs(
    Pint        ar_id,                 /* archive identifier                 */
    Pint        num_elems_appl_list,   /* # elems in appl list               */
    Pint        start_ind,             /* starting index                     */
    Pint        *err_ind,              /* OUT error indicator                */
    Pint_list   *ids,                  /* OUT list of conflicting structure ids */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE CONFLICTING STRUCTURES IN NETWORK */
extern void pinq_conf_structs_net(
    Pint                ar_id,                /* archive identifier              */
    Pint                struct_id,            /* structure identifier            */
    Pstruct_net_source  source,               /* structure network source        */
    Pint                num_elems_appl_list,  /* # elems in appl list            */
    Pint                start_ind,            /* starting index                  */
    Pint                *err_ind,             /* OUT error indicator             */
    Pint_list           *ids,                 /* OUT conflicting struct id list  */
    Pint                *num_elems_impl_list  /* OUT # elems in impl list        */
);

/* INQUIRE MORE SIMULTANEOUS EVENTS */
extern void pinq_more_simult_events(
    Pint                 *err_ind,        /* OUT error indicator      */
    Pmore_simult_events  *simult_events   /* OUT simultaneous events  */
);

/* INQUIRE WORKSTATION CONNECTION AND TYPE */
extern void pinq_ws_conn_type(
    Pint    ws_id,       /* workstation identifier     */
    Pstore  store,       /* handle to Store object     */
    Pint    *err_ind,    /* OUT error indicator        */
    void    **conn_id,   /* OUT connection identifier  */
    Pint    *ws_type     /* OUT workstation type       */
);

/* INQUIRE LIST OF VIEW INDICES */
extern void pinq_list_view_inds(
    Pint       ws_id,                /* workstation identifier    */
    Pint       num_elems_appl_list,  /* # elems in appl list      */
    Pint       start_ind,            /* starting index            */
    Pint       *err_ind,             /* OUT error indicator       */
    Pint_list  *view_inds,           /* OUT list of view indices  */
    Pint       *num_elems_impl_list  /* OUT # elems in impl list  */
);

/* INQUIRE VIEW REPRESENTATION */
extern void pinq_view_rep(
    Pint         ws_id,      /* workstation identifier            */
    Pint         view_ind,   /* view index                        */
    Pint         *err_ind,   /* OUT error indicator               */
    Pupd_st      *upd_st,    /* OUT transformation update state   */
    Pview_rep3   *cur_view,  /* OUT current view representation   */
    Pview_rep3   *req_view   /* OUT requested view representation */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE HLHSR MODE */
extern void pinq_hlhsr_mode(
    Pint      ws_id,       /* workstation identifier     */
    Pint      *err_ind,    /* OUT error indicator        */
    Pupd_st   *upd_st,     /* OUT HLHSR update state     */
    Pint      *cur_mode,   /* OUT current HLHSR mode     */
    Pint      *req_mode    /* OUT requested HLHSR mode   */
);

/* INQUIRE POSTED STRUCTURES */
extern void pinq_posted_structs(
    Pint                ws_id,                /* workstation identifier          */
    Pint                num_elems_appl_list,  /* # elems in appl list            */
    Pint                start_ind,            /* starting index                  */
    Pint                *err_ind,             /* OUT error indicator             */
    Pposted_struct_list *struct_ids,          /* OUT list of posted structures   */
    Pint                *num_elems_impl_list  /* OUT # elems in impl list        */
);

/* INQUIRE DISPLAY UPDATE STATE */
extern void pinq_disp_upd_st(
    Pint              ws_id,             /* workstation identifier              */
    Pint              *err_ind,          /* OUT error indicator                 */
    Pdefer_mode       *def_mode,         /* OUT deferral mode                   */
    Pmod_mode         *mod_mode,         /* OUT modification mode               */
    Pdisp_surf_empty  *disp_surf_empty,  /* OUT display surface empty           */
    Pvisual_st        *vis_st            /* OUT state of visual representation  */
);

/* INQUIRE LIST OF POLYLINE INDICES */
extern void pinq_list_line_inds(
    Pint       ws_id,                /* workstation identifier                */
    Pint       num_elems_appl_list,  /* # elems in appl list                  */
    Pint       start_ind,            /* starting index                        */
    Pint       *err_ind,             /* OUT error indicator                   */
    Pint_list  *def_line_ind,        /* OUT list of defined polyline indices  */
    Pint       *num_elems_impl_list  /* OUT # elems in impl list              */
);

/* INQUIRE POLYLINE REPRESENTATION */
extern void pinq_line_rep(
    Pint          ws_id,      /* workstation identifier       */
    Pint          index,      /* polyline index               */
    Pinq_type     type,       /* type of returned value       */
    Pint          *err_ind,   /* OUT error indicator          */
    Pline_bundle  *line_rep   /* OUT polyline representation  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE LIST OF POLYMARKER INDICES */
extern void pinq_list_marker_inds(
    Pint        ws_id,                  /* workstation identifier                  */
    Pint        num_elems_appl_list,    /* # elems in appl list                    */
    Pint        start_ind,              /* starting index                          */
    Pint        *err_ind,               /* OUT error indicator                     */
    Pint_list   *def_marker_ind,        /* OUT list of defined polymarker indices  */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list                */
);

/* INQUIRE POLYMARKER REPRESENTATION */
extern void pinq_marker_rep(
    Pint            ws_id,          /* workstation identifier          */
    Pint            index,          /* polymarker index                */
    Pinq_type       type,           /* type of returned value          */
    Pint            *err_ind,       /* OUT error indicator             */
    Pmarker_bundle  *marker_rep     /* OUT polymarker representation   */
);

/* INQUIRE LIST OF TEXT INDICES */
extern void pinq_list_text_inds(
    Pint        ws_id,                  /* workstation identifier            */
    Pint        num_elems_appl_list,    /* # elems in appl list              */
    Pint        start_ind,              /* starting index                    */
    Pint        *err_ind,               /* OUT error indicator               */
    Pint_list   *def_text_ind,          /* OUT list of defined text indices  */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list          */
);

/* INQUIRE TEXT REPRESENTATION */
extern void pinq_text_rep(
    Pint          ws_id,        /* workstation identifier   */
    Pint          index,        /* text index               */
    Pinq_type     type,         /* type of returned value   */
    Pint          *err_ind,     /* OUT error indicator      */
    Ptext_bundle  *text_rep     /* OUT text representation  */
);

/* INQUIRE LIST OF INTERIOR INDICES */
extern void pinq_list_int_inds(
    Pint        ws_id,                  /* workstation identifier                */
    Pint        num_elems_appl_list,    /* # elems in appl list                  */
    Pint        start_ind,              /* starting index                        */
    Pint        *err_ind,               /* OUT error indicator                   */
    Pint_list   *def_int_ind,           /* OUT list of defined interior indices  */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list              */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577562 489 ■


```
/* INQUIRE INTERIOR REPRESENTATION */
extern void pinq_int_rep(
    Pint          ws_id,      /* workstation identifier       */
    Pint          index,      /* interior index               */
    Pinq_type     type,       /* type of returned value       */
    Pint          *err_ind,   /* OUT error indicator          */
    Pint_bundle   *int_rep    /* OUT interior representation  */
);

/* INQUIRE LIST OF EDGE INDICES */
extern void pinq_list_edge_inds(
    Pint        ws_id,                /* workstation identifier            */
    Pint        num_elems_appl_list,  /* # elems in appl list              */
    Pint        start_ind,            /* starting index                    */
    Pint        *err_ind,             /* OUT error indicator               */
    Pint_list   *def_edge_ind,        /* OUT list of defined edge indices  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list          */
);

/* INQUIRE EDGE REPRESENTATION */
extern void pinq_edge_rep(
    Pint          ws_id,      /* workstation identifier   */
    Pint          index,      /* edge index               */
    Pinq_type     type,       /* type of returned value   */
    Pint          *err_ind,   /* OUT error indicator      */
    Pedge_bundle  *edge_rep   /* OUT edge representation  */
);

/* INQUIRE LIST OF PATTERN INDICES */
extern void pinq_list_pat_inds(
    Pint        ws_id,                /* workstation identifier               */
    Pint        num_elems_appl_list,  /* # elems in appl list                 */
    Pint        start_ind,            /* starting index                       */
    Pint        *err_ind,             /* OUT error indicator                  */
    Pint_list   *def_pat_ind,         /* OUT list of defined pattern indices  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list             */
);

/* INQUIRE PATTERN REPRESENTATION */
extern void pinq_pat_rep(
    Pint        ws_id,      /* workstation identifier      */
    Pint        index,      /* pattern index               */
    Pinq_type   type,       /* type of returned value      */
    Pstore      store,      /* handle to Store object      */
    Pint        *err_ind,   /* OUT error indicator         */
    Ppat_rep    **pat_rep   /* OUT pattern representation  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE COLOUR MODEL */
extern void pinq_colr_model(
    Pint   ws_id,      /* workstation identifier     */
    Pint   *err_ind,   /* OUT error indicator        */
    Pint   *model      /* OUT current colour model   */
);

/* INQUIRE LIST OF COLOUR INDICES */
extern void pinq_list_colr_inds(
    Pint       ws_id,                /* workstation identifier       */
    Pint       num_elems_appl_list,  /* # elems in appl list         */
    Pint       start_ind,            /* starting index               */
    Pint       *err_ind,             /* OUT error indicator          */
    Pint_list  *colr_ind,            /* OUT list of colour indices   */
    Pint       *num_elems_impl_list  /* OUT # elems in impl list     */
);

/* INQUIRE COLOUR REPRESENTATION */
extern void pinq_colr_rep(
    Pint       ws_id,      /* workstation identifier     */
    Pint       colr_ind,   /* colour index               */
    Pinq_type  type,       /* type of returned value     */
    Pint       *err_ind,   /* OUT error indicator        */
    Pcolr_rep  *colr_rep   /* OUT colour representation   */
);

/* INQUIRE HIGHLIGHTING FILTER */
extern void pinq_highl_filter(
    Pint     ws_id,          /* workstation identifier   */
    Pstore   store,          /* handle to Store object   */
    Pint     *err_ind,       /* OUT error indicator      */
    Pfilter  **highl_filter  /* OUT highlighting filter  */
);

/* INQUIRE INVISIBILITY FILTER */
extern void pinq_invis_filter(
    Pint     ws_id,          /* workstation identifier   */
    Pstore   store,          /* handle to Store object   */
    Pint     *err_ind,       /* OUT error indicator      */
    Pfilter  **invis_filter  /* OUT invisibility filter  */
);

/* INQUIRE WORKSTATION TRANSFORMATION 3 */
extern void pinq_ws_tran3(
    Pint     ws_id,          /* workstation identifier              */
    Pint     *err_ind,       /* OUT error indicator                 */
    Pupd_st  *upd_st,        /* OUT update state                    */
    Plimit3  *req_win_lim,   /* OUT requested workstation window    */
    Plimit3  *cur_win_lim,   /* OUT current workstation window      */
    Plimit3  *req_vp_lim,    /* OUT requested workstation viewport  */
    Plimit3  *cur_vp_lim     /* OUT current workstation viewport    */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE WORKSTATION TRANSFORMATION */
extern void pinq_ws_tran(
    Pint      ws_id,          /* workstation identifier              */
    Pint      *err_ind,       /* OUT error indicator                 */
    Pupd_st   *upd_st,        /* OUT update state                    */
    Plimit    *req_win_lim,   /* OUT requested workstation window    */
    Plimit    *cur_win_lim,   /* OUT current workstation window      */
    Plimit    *req_vp_lim,    /* OUT requested workstation viewport  */
    Plimit    *cur_vp_lim     /* OUT current workstation viewport    */
);

/* INQUIRE LOCATOR DEVICE STATE 3 */
extern void pinq_loc_st3(
    Pint          ws_id,          /* workstation identifier        */
    Pint          loc_num,        /* locator device number         */
    Pinq_type     type,           /* type of returned value        */
    Pstore        store,          /* handle to Store object        */
    Pint          *err_ind,       /* OUT error indicator           */
    Pop_mode      *op_mode,       /* OUT operating mode            */
    Pecho_switch  *echo_switch,   /* OUT echo switch               */
    Pint          *init_view_ind, /* OUT initial view index        */
    Ppoint3       *init_loc_pos,  /* OUT initial locator position  */
    Pint          *prompt_echo,   /* OUT prompt/echo type          */
    Plimit3       *echo_vol,      /* OUT echo volume               */
    Ploc_data3    **loc_data      /* OUT data record               */
);

/* INQUIRE LOCATOR DEVICE STATE */
extern void pinq_loc_st(
    Pint          ws_id,          /* workstation identifier        */
    Pint          loc_num,        /* locator device number         */
    Pinq_type     type,           /* type of returned value        */
    Pstore        store,          /* handle to Store object        */
    Pint          *err_ind,       /* OUT error indicator           */
    Pop_mode      *op_mode,       /* OUT operating mode            */
    Pecho_switch  *echo_switch,   /* OUT echo switch               */
    Pint          *init_view_ind, /* OUT initial view index        */
    Ppoint        *init_loc_pos,  /* OUT initial locator position  */
    Pint          *prompt_echo,   /* OUT prompt/echo type          */
    Plimit        *echo_area,     /* OUT echo area                 */
    Ploc_data     **loc_data      /* OUT data record               */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE STROKE DEVICE STATE 3 */
extern void pinq_stroke_st3(
    Pint            ws_id,          /* workstation identifier   */
    Pint            stroke_num,     /* stroke device number     */
    Pinq_type       type,           /* type of returned value   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pint            *init_view_ind, /* OUT initial view index   */
    Ppoint_list3    **init_stroke,  /* OUT initial stroke       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit3         *echo_vol,      /* OUT echo volume          */
    Pstroke_data3   **stroke_data   /* OUT data record          */
);

/* INQUIRE STROKE DEVICE STATE */
extern void pinq_stroke_st(
    Pint            ws_id,          /* workstation identifier   */
    Pint            stroke_num,     /* stoke device number      */
    Pinq_type       type,           /* type of returned value   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pint            *init_view_ind, /* OUT initial view index   */
    Ppoint_list     **init_stroke,  /* OUT initial stroke       */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit          *echo_area,     /* OUT echo area            */
    Pstroke_data    **stroke_data   /* OUT data record          */
);

/* INQUIRE VALUATOR DEVICE STATE 3 */
extern void pinq_val_st3(
    Pint            ws_id,          /* workstation identifier   */
    Pint            val_num,        /* valuator device number   */
    Pstore          store,          /* handle to Store object   */
    Pint            *err_ind,       /* OUT error indicator      */
    Pop_mode        *op_mode,       /* OUT operating mode       */
    Pecho_switch    *echo_switch,   /* OUT echo switch          */
    Pfloat          *init_value,    /* OUT initial value        */
    Pint            *prompt_echo,   /* OUT prompt/echo type     */
    Plimit3         *echo_vol,      /* OUT echo volume          */
    Pval_data3      **val_data      /* OUT data record          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE VALUATOR DEVICE STATE */
extern void pinq_val_st(
    Pint          ws_id,          /* workstation identifier   */
    Pint          val_num,        /* valuator device number   */
    Pstore        store,          /* handle to Store object   */
    Pint          *err_ind,       /* OUT error indicator      */
    Pop_mode      *op_mode,       /* OUT operating mode       */
    Pecho_switch  *echo_switch,   /* OUT echo switch          */
    Pfloat        *init_value,    /* OUT initial value        */
    Pint          *prompt_echo,   /* OUT prompt/echo type     */
    Plimit        *echo_area,     /* OUT echo area            */
    Pval_data     **val_data      /* OUT data record          */
);

/* INQUIRE CHOICE DEVICE STATE 3 */
extern void pinq_choice_st3(
    Pint           ws_id,          /* workstation identifier     */
    Pint           choice_num,     /* choice device number       */
    Pstore         store,          /* handle to Store object     */
    Pint           *err_ind,       /* OUT error indicator        */
    Pop_mode       *op_mode,       /* OUT operating mode         */
    Pecho_switch   *echo_switch,   /* OUT echo switch            */
    Pin_status     *init_status,   /* OUT initial choice status  */
    Pint           *init_choice,   /* OUT initial choice         */
    Pint           *prompt_echo,   /* OUT prompt/echo type       */
    Plimit3        *echo_vol,      /* OUT echo volume            */
    Pchoice_data3  **choice_data   /* OUT data record            */
);

/* INQUIRE CHOICE DEVICE STATE */
extern void pinq_choice_st(
    Pint          ws_id,          /* workstation identifier     */
    Pint          choice_num,     /* choice device number       */
    Pstore        store,          /* handle to Store object     */
    Pint          *err_ind,       /* OUT error indicator        */
    Pop_mode      *op_mode,       /* OUT operating mode         */
    Pecho_switch  *echo_switch,   /* OUT echo switch            */
    Pin_status    *init_status,   /* OUT initial choice status  */
    Pint          *init_choice,   /* OUT initial choice         */
    Pint          *prompt_echo,   /* OUT prompt/echo type       */
    Plimit        *echo_area,     /* OUT echo area              */
    Pchoice_data  **choice_data   /* OUT data record            */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PICK DEVICE STATE 3 */
extern void pinq_pick_st3(
    Pint           ws_id,          /* workstation identifier    */
    Pint           pick_num,       /* pick device number        */
    Pinq_type      type,           /* type of returned value    */
    Pstore         store,          /* handle to Store object    */
    Pint           *err_ind,       /* OUT error indicator       */
    Pop_mode       *op_mode,       /* OUT operating mode        */
    Pecho_switch   *echo_switch,   /* OUT echo switch           */
    Pfilter        **pick_filter,  /* OUT pick filter           */
    Pin_status     *init_status,   /* OUT initial pick status   */
    Ppick_path     **init_pick,    /* OUT initial pick path     */
    Pint           *prompt_echo,   /* OUT prompt/echo type      */
    Plimit3        *echo_vol,      /* OUT echo volume           */
    Ppick_data3    **pick_data,    /* OUT data record           */
    Ppath_order    *order          /* OUT pick path order       */
);

/* INQUIRE PICK DEVICE STATE */
extern void pinq_pick_st(
    Pint           ws_id,          /* workstation identifier    */
    Pint           pick_num,       /* pick device number        */
    Pinq_type      type,           /* type of returned value    */
    Pstore         store,          /* handle to Store object    */
    Pint           *err_ind,       /* OUT error indicator       */
    Pop_mode       *op_mode,       /* OUT operating mode        */
    Pecho_switch   *echo_switch,   /* OUT echo switch           */
    Pfilter        **pick_filter,  /* OUT pick filter           */
    Pin_status     *init_status,   /* OUT initial pick status   */
    Ppick_path     **init_pick,    /* OUT initial pick path     */
    Pint           *prompt_echo,   /* OUT prompt/echo type      */
    Plimit         *echo_area,     /* OUT echo area             */
    Ppick_data     **pick_data,    /* OUT data record           */
    Ppath_order    *order          /* OUT pick path order       */
);

/* INQUIRE STRING DEVICE STATE 3 */
extern void pinq_string_st3(
    Pint           ws_id,          /* workstation identifier    */
    Pint           string_num,     /* string device number      */
    Pstore         store,          /* handle to Store object    */
    Pint           *err_ind,       /* OUT error indicator       */
    Pop_mode       *op_mode,       /* OUT operating mode        */
    Pecho_switch   *echo_switch,   /* OUT echo switch           */
    char           **init_string,  /* OUT initial string        */
    Pint           *prompt_echo,   /* OUT prompt/echo type      */
    Plimit3        *echo_vol,      /* OUT echo volume           */
    Pstring_data3  **string_data   /* OUT data record           */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE STRING DEVICE STATE */
extern void pinq_string_st(
    Pint          ws_id,          /* workstation identifier   */
    Pint          string_num,     /* string device number     */
    Pstore        store,          /* handle to Store object   */
    Pint          *err_ind,       /* OUT error indicator      */
    Pop_mode      *op_mode,       /* OUT operating mode       */
    Pecho_switch  *echo_switch,   /* OUT echo switch          */
    char          **init_string,  /* OUT initial string       */
    Pint          *prompt_echo,   /* OUT prompt/echo type     */
    Plimit        *echo_area,     /* OUT echo area            */
    Pstring_data  **string_data   /* OUT data record          */
);

/* INQUIRE WORKSTATION CATEGORY */
extern void pinq_ws_cat(
    Pint     ws_type,    /* workstation type          */
    Pint     *err_ind,   /* OUT error indicator       */
    Pws_cat  *cat        /* OUT workstation category  */
);

/* INQUIRE DISPLAY SPACE SIZE 3 */
extern void pinq_disp_space_size3(
    Pint               ws_type,    /* workstation type      */
    Pint               *err_ind,   /* OUT error indicator   */
    Pdisp_space_size3  *size       /* OUT display size      */
);

/* INQUIRE DISPLAY SPACE SIZE */
extern void pinq_disp_space_size(
    Pint              ws_type,    /* workstation type      */
    Pint              *err_ind,   /* OUT error indicator   */
    Pdisp_space_size  *size       /* OUT display size      */
);

/* INQUIRE HLHSR IDENTIFIER FACILITIES */
extern void pinq_hlhsr_id_facs(
    Pint       ws_type,              /* workstation type          */
    Pint       num_elems_appl_list,  /* # elems in appl list      */
    Pint       start_ind,            /* starting index            */
    Pint       *err_ind,             /* OUT error indicator       */
    Pint_list  *hlhsr_ids,           /* OUT list of HLHSR ids     */
    Pint       *num_elems_impl_list  /* OUT # elems in impl list  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE HLHSR MODE FACILITIES */
extern void pinq_hlhsr_mode_facs(
    Pint        ws_type,               /* workstation type          */
    Pint        num_elems_appl_list,   /* # elems in appl list      */
    Pint        start_ind,             /* starting index            */
    Pint        *err_ind,              /* OUT error indicator       */
    Pint_list   *hlhsr_modes,          /* OUT list of HLHSR modes   */
    Pint        *num_elems_impl_list   /* OUT # elems in impl list  */
);

/* INQUIRE VIEW FACILITIES */
extern void pinq_view_facs(
    Pint   ws_type,        /* workstation type                      */
    Pint   *err_ind,       /* OUT error indicator                   */
    Pint   *num_view_ind   /* OUT number of predefined view indices */
);

/* INQUIRE PREDEFINED VIEW REPRESENTATION */
extern void pinq_pred_view_rep(
    Pint         ws_type,    /* workstation type         */
    Pint         index,      /* predefined view index    */
    Pint         *err_ind,   /* OUT error indicator      */
    Pview_rep3   *view       /* OUT view representation  */
);

/* INQUIRE WORKSTATION CLASSIFICATION */
extern void pinq_ws_class(
    Pint        ws_type,    /* workstation type       */
    Pint        *err_ind,   /* OUT error indicator    */
    Pws_class   *ws_class   /* OUT workstation class  */
);

/* INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES */
extern void pinq_dyns_ws_attrs(
    Pint             ws_type,    /* workstation type          */
    Pint             *err_ind,   /* OUT error indicator       */
    Pdyns_ws_attrs   *attr       /* OUT attributes dynamics   */
);

/* INQUIRE DEFAULT DISPLAY UPDATE STATE */
extern void pinq_def_disp_upd_st(
    Pint          ws_type,     /* workstation type       */
    Pint          *err_ind,    /* OUT error indicator    */
    Pdefer_mode   *def_mode,   /* OUT deferral mode      */
    Pmod_mode     *mod_mode    /* OUT modification mode  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE POLYLINE FACILITIES */
extern void pinq_line_facs(
    Pint         ws_type,               /* workstation type          */
    Pint         num_elems_appl_list,   /* # elems in appl list      */
    Pint         start_ind,             /* starting index            */
    Pint         *err_ind,              /* OUT error indicator       */
    Pline_facs   *fac,                  /* OUT polyline facilities   */
    Pint         *num_elems_impl_list   /* OUT # elems in impl list  */
);

/* INQUIRE PREDEFINED POLYLINE REPRESENTATION */
extern void pinq_pred_line_rep(
    Pint          ws_type,    /* workstation type             */
    Pint          index,      /* predefined index             */
    Pint          *err_ind,   /* OUT error indicator          */
    Pline_bundle  *bundle     /* OUT predefined polyline rep  */
);

/* INQUIRE POLYMARKER FACILITIES */
extern void pinq_marker_facs(
    Pint          ws_type,               /* workstation type          */
    Pint          num_elems_appl_list,   /* # elems in appl list      */
    Pint          start_ind,             /* starting index            */
    Pint          *err_ind,              /* OUT error indicator       */
    Pmarker_facs  *fac,                  /* OUT polymarker facilities */
    Pint          *num_elems_impl_list   /* OUT # elems in impl list  */
);

/* INQUIRE PREDEFINED POLYMARKER REPRESENTATION */
extern void pinq_pred_marker_rep(
    Pint             ws_type,    /* workstation type              */
    Pint             index,      /* predefined index              */
    Pint             *err_ind,   /* OUT error indicator           */
    Pmarker_bundle   *bundle     /* OUT predefined polymarker rep */
);

/* INQUIRE TEXT FACILITIES */
extern void pinq_text_facs(
    Pint         ws_type,               /* workstation type          */
    Pint         num_elems_appl_list,   /* # elems in appl list      */
    Pint         start_ind,             /* starting index            */
    Pint         *err_ind,              /* OUT error indicator       */
    Ptext_facs   *fac,                  /* OUT text facilities       */
    Pint         *num_elems_impl_list   /* OUT # elems in impl list  */
);

/* INQUIRE PREDEFINED TEXT REPRESENTATION */
extern void pinq_pred_text_rep(
    Pint          ws_type,    /* workstation type          */
    Pint          index,      /* predefined index          */
    Pint          *err_ind,   /* OUT error indicator       */
    Ptext_bundle  *bundle     /* OUT predefined text rep   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE ANNOTATION FACILITIES */
extern void pinq_anno_facs(
    Pint        ws_type,                /* workstation type                          */
    Pint        num_elems_appl_list,    /* # elems in appl list                      */
    Pint        start_ind,              /* starting index                            */
    Pint        *err_ind,               /* OUT error indicator                       */
    Pint_list   *styles,                /* OUT list annotation styles                */
    Pint        *num_elems_impl_list,   /* OUT # elems in impl list                  */
    Pint        *num_anno_char_hts,     /* OUT number of anno. text char. heights    */
    Pfloat      *min_anno_char_ht,      /* OUT minimum anno. text char. height       */
    Pfloat      *max_anno_char_ht       /* OUT maximum anno. text char. height       */
);

/* INQUIRE TEXT EXTENT */
extern void pinq_text_extent(
    Pint              ws_type,          /* workstation type           */
    Pint              text_font,        /* text font                  */
    Pfloat            char_expan,       /* char expansion factor      */
    Pfloat            char_space,       /* char spacing               */
    Pfloat            char_ht,          /* char height                */
    Ptext_path        text_path,        /* text path                  */
    Phor_text_align   hor_text_align,   /* horizontal text alignment  */
    Pvert_text_align  ver_text_align,   /* vertical text alignment    */
    const char        *char_string,     /* character string           */
    Pint              *err_ind,         /* OUT error indicator        */
    Prect             *rect,            /* OUT extent rectangle       */
    Ppoint            *offset           /* OUT concatenation offset   */
);

/* INQUIRE INTERIOR FACILITIES */
extern void pinq_int_facs(
    Pint        ws_type,                        /* workstation type                    */
    Pint        hatch_num_elems_appl_length,    /* # elems in appl hatch style list    */
    Pint        hatch_start_ind,                /* starting index                      */
    Pint        *err_ind,                       /* OUT error indicator                 */
    Pint_facs   *int_facs,                      /* OUT interior facilities             */
    Pint        *hatch_num_elems_impl_list      /* OUT # elems in impl list            */
);

/* INQUIRE PREDEFINED INTERIOR REPRESENTATION */
extern void pinq_pred_int_rep(
    Pint          ws_type,    /* workstation type              */
    Pint          index,      /* predefined index              */
    Pint          *err_ind,   /* OUT error indicator           */
    Pint_bundle   *bundle     /* OUT predefined interior rep   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE EDGE FACILITIES */
extern void pinq_edge_facs(
    Pint        ws_type,              /* workstation type          */
    Pint        num_elems_appl_list,  /* # elems in appl list      */
    Pint        start_ind,            /* starting index            */
    Pint        *err_ind,             /* OUT error indicator       */
    Pedge_facs  *fac,                 /* OUT edge facilities       */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list  */
);

/* INQUIRE PREDEFINED EDGE REPRESENTATION */
extern void pinq_pred_edge_rep(
    Pint          ws_type,    /* workstation type          */
    Pint          index,      /* predefined index          */
    Pint          *err_ind,   /* OUT error indicator       */
    Pedge_bundle  *bundle     /* OUT predefined edge rep   */
);

/* INQUIRE PATTERN FACILITIES */
extern void pinq_pat_facs(
    Pint  ws_type,    /* workstation type                          */
    Pint  *err_ind,   /* OUT error indicator                       */
    Pint  *num_pred   /* OUT number of predefined pattern indices  */
);

/* INQUIRE PREDEFINED PATTERN REPRESENTATION */
extern void pinq_pred_pat_rep(
    Pint      ws_type,    /* workstation type          */
    Pint      index,      /* predefined index          */
    Pstore    store,      /* handle to Store object    */
    Pint      *err_ind,   /* OUT error indicator       */
    Ppat_rep  **pat_rep   /* OUT predefined pattern rep */
);

/* INQUIRE COLOUR MODEL FACILITIES */
extern void pinq_colr_model_facs(
    Pint        ws_type,              /* workstation type          */
    Pint        num_elems_appl_list,  /* # elems in appl list      */
    Pint        start_ind,            /* starting index            */
    Pint        *err_ind,             /* OUT error indicator       */
    Pint_list   *models,              /* OUT list of colour models */
    Pint        *num_elems_impl_list, /* OUT # elems in impl list  */
    Pint        *def                  /* OUT default colour model  */
);

/* INQUIRE COLOUR FACILITIES */
extern void pinq_colr_facs(
    Pint        ws_type,    /* workstation type       */
    Pint        *err_ind,   /* OUT error indicator    */
    Pcolr_facs  *fac        /* OUT colour facilities  */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PREDEFINED COLOUR REPRESENTATION */
extern void pinq_pred_colr_rep(
    Pint        ws_type,    /* workstation type                        */
    Pint        colr_ind,   /* predefined index                        */
    Pint        *err_ind,   /* OUT error indicator                     */
    Pcolr_rep   *colr_rep   /* OUT predefined colour representation    */
);

/* INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3 */
extern void pinq_list_avail_gdp3(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *gdps,                  /* OUT list of GDPs          */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);

/* INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES */
extern void pinq_list_avail_gdp(
    Pint        ws_type,                /* workstation type          */
    Pint        num_elems_appl_list,    /* # elems in appl list      */
    Pint        start_ind,              /* starting index            */
    Pint        *err_ind,               /* OUT error indicator       */
    Pint_list   *gdps,                  /* OUT list of GDPs          */
    Pint        *num_elems_impl_list    /* OUT # elems in impl list  */
);

/* INQUIRE GENERALIZED DRAWING PRIMITIVE 3 */
extern void pinq_gdp3(
    Pint    ws_type,    /* workstation type                 */
    Pint    gdp,        /* GDP function identifier          */
    Pint    *err_ind,   /* OUT error indicator              */
    Pint    *num_attr,  /* OUT number of attributes used    */
    Pattrs  attr[5]     /* OUT list of attributes used      */
);

/* INQUIRE GENERALIZED DRAWING PRIMITIVE */
extern void pinq_gdp(
    Pint    ws_type,    /* workstation type                 */
    Pint    gdp,        /* GDP function identifier          */
    Pint    *err_ind,   /* OUT error indicator              */
    Pint    *num_attr,  /* OUT number of attributes used    */
    Pattrs  attr[5]     /* OUT list of attributes used      */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS */
extern void pinq_list_avail_gse(
    Pint        ws_type,              /* workstation type          */
    Pint        num_elems_appl_list,  /* # elems in appl list      */
    Pint        start_ind,            /* starting index            */
    Pint        *err_ind,             /* OUT error indicator       */
    Pint_list   *gses,                /* OUT list of GSEs          */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list  */
);

/* INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED */
extern void pinq_num_disp_pris(
    Pint   ws_type,    /* workstation type                    */
    Pint   *err_ind,   /* OUT error indicator                 */
    Pint   *num_pri    /* OUT number of structure priorities  */
);

/* INQUIRE WORKSTATION STATE TABLE LENGTHS */
extern void pinq_ws_st_table(
    Pint            ws_type,    /* workstation type                  */
    Pint            *err_ind,   /* OUT error indicator               */
    Pws_st_tables   *lengths    /* OUT lengths of workstation tables */
);

/* INQUIRE DYNAMICS OF STRUCTURES */
extern void pinq_dyns_structs(
    Pint            ws_type,    /* workstation type       */
    Pint            *err_ind,   /* OUT error indicator    */
    Pdyns_structs   *dyns       /* OUT structure dynamics */
);

/* INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES */
extern void pinq_num_avail_in(
    Pint      ws_type,    /* workstation type           */
    Pint      *err_ind,   /* OUT error indicator        */
    Pnum_in   *num_in     /* OUT number of input devices */
);

/* INQUIRE DEFAULT LOCATOR DEVICE DATA 3 */
extern void pinq_def_loc_data3(
    Pint         ws_type,      /* workstation type                 */
    Pint         loc_num,      /* logical input device number      */
    Pstore       store,        /* handle to Store object           */
    Pint         *err_ind,     /* OUT error indicator              */
    Ppoint3      *loc_pos,     /* OUT default initial position     */
    Pint_list    **pet_list,   /* OUT list of prompt and echo types */
    Plimit3      *echo_vol,    /* OUT default echo volume          */
    Ploc_data3   **loc_data    /* OUT default data record          */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEFAULT LOCATOR DEVICE DATA */
extern void pinq_def_loc_data(
    Pint        ws_type,      /* workstation type                    */
    Pint        loc_num,      /* logical input device number         */
    Pstore      store,        /* handle to Store object              */
    Pint        *err_ind,     /* OUT error indicator                 */
    Ppoint      *loc_pos,     /* OUT default locator position        */
    Pint_list   **pet_list,   /* OUT list of prompt and echo types   */
    Plimit      *echo_area,   /* OUT default echo area               */
    Ploc_data   **loc_data    /* OUT default data record             */
);

/* INQUIRE DEFAULT STROKE DEVICE DATA 3 */
extern void pinq_def_stroke_data3(
    Pint            ws_type,         /* workstation type                    */
    Pint            stroke_num,      /* logical input device number         */
    Pstore          store,           /* handle to Store object              */
    Pint            *err_ind,        /* OUT error indicator                 */
    Pint            *max_buf_size,   /* OUT max. input buffer size          */
    Pint_list       **pet_list,      /* OUT list of prompt and echo types   */
    Plimit3         *echo_vol,       /* OUT default echo volume             */
    Pstroke_data3   **stroke_data    /* OUT default data record             */
);

/* INQUIRE DEFAULT STROKE DEVICE DATA */
extern void pinq_def_stroke_data(
    Pint            ws_type,         /* workstation type                    */
    Pint            stroke_num,      /* logical input device number         */
    Pstore          store,           /* handle to Store object              */
    Pint            *err_ind,        /* OUT error indicator                 */
    Pint            *max_buf_size,   /* OUT max. input buffer size          */
    Pint_list       **pet_list,      /* OUT list of prompt and echo types   */
    Plimit          *echo_area,      /* OUT default echo area               */
    Pstroke_data    **stroke_data    /* OUT default data record             */
);

/* INQUIRE DEFAULT VALUATOR DEVICE DATA 3 */
extern void pinq_def_val_data3(
    Pint        ws_type,      /* workstation type                    */
    Pint        val_num,      /* logical input device number         */
    Pstore      store,        /* handle to Store object              */
    Pint        *err_ind,     /* OUT error indicator                 */
    Pfloat      *def_value,   /* OUT default initial value           */
    Pint_list   **pet_list,   /* OUT list of prompt and echo types   */
    Plimit3     *echo_vol,    /* OUT default echo volume             */
    Pval_data3  **val_data    /* OUT default data record             */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE DEFAULT VALUATOR DEVICE DATA */
extern void pinq_def_val_data(
    Pint        ws_type,      /* workstation type                    */
    Pint        val_num,      /* logical input device number         */
    Pstore      store,        /* handle to Store object              */
    Pint        *err_ind,     /* OUT error indicator                 */
    Pfloat      *def_value,   /* OUT default initial value           */
    Pint_list   **pet_list,   /* OUT list of prompt and echo types   */
    Plimit      *echo_area,   /* OUT default echo area               */
    Pval_data   **val_data    /* OUT default data record             */
);

/* INQUIRE DEFAULT CHOICE DEVICE DATA 3 */
extern void pinq_def_choice_data3(
    Pint            ws_type,        /* workstation type                    */
    Pint            choice_num,     /* logical input device number         */
    Pstore          store,          /* handle to Store object              */
    Pint            *err_ind,       /* OUT error indicator                 */
    Pint            *max_choices,   /* OUT max. number of choices          */
    Pint_list       **pet_list,     /* OUT list of prompt and echo types   */
    Plimit3         *echo_vol,      /* OUT default echo volume             */
    Pchoice_data3   **choice_data   /* OUT default data record             */
);

/* INQUIRE DEFAULT CHOICE DEVICE DATA */
extern void pinq_def_choice_data(
    Pint            ws_type,        /* workstation type                    */
    Pint            choice_num,     /* logical input device number         */
    Pstore          store,          /* handle to Store object              */
    Pint            *err_ind,       /* OUT error indicator                 */
    Pint            *max_choices,   /* OUT max. number of choices          */
    Pint_list       **pet_list,     /* OUT list of prompt and echo types   */
    Plimit          *echo_area,     /* OUT default echo area               */
    Pchoice_data    **choice_data   /* OUT default data record             */
);

/* INQUIRE DEFAULT PICK DEVICE DATA 3 */
extern void pinq_def_pick_data3(
    Pint          ws_type,      /* workstation type                    */
    Pint          pick_num,     /* logical input device number         */
    Pstore        store,        /* handle to Store object              */
    Pint          *err_ind,     /* OUT error indicator                 */
    Pint_list     **pet_list,   /* OUT list of prompt and echo types   */
    Plimit3       *echo_vol,    /* OUT default echo volume             */
    Ppick_data3   **pick_data   /* OUT default data record             */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE DEFAULT PICK DEVICE DATA */
extern void pinq_def_pick_data(
    Pint         ws_type,      /* workstation type                  */
    Pint         pick_num,     /* logical input device number       */
    Pstore       store,        /* handle to Store object            */
    Pint         *err_ind,     /* OUT error indicator               */
    Pint_list    **pet_list,   /* OUT list of prompt and echo types */
    Plimit       *echo_area,   /* OUT default echo area             */
    Ppick_data   **pick_data   /* OUT default data record           */
);

/* INQUIRE DEFAULT STRING DEVICE DATA 3 */
extern void pinq_def_string_data3(
    Pint            ws_type,         /* workstation type                  */
    Pint            string_num,      /* logical input device number       */
    Pstore          store,           /* handle to Store object            */
    Pint            *err_ind,        /* OUT error indicator               */
    Pint            *max_buf_size,   /* OUT max. input buffer size        */
    Pint_list       **pet_list,      /* OUT list of prompt and echo types */
    Plimit3         *echo_vol,       /* OUT default echo volume           */
    Pstring_data3   **string_data    /* OUT default data record           */
);

/* INQUIRE DEFAULT STRING DEVICE DATA */
extern void pinq_def_string_data(
    Pint           ws_type,         /* workstation type                  */
    Pint           string_num,      /* logical input device number       */
    Pstore         store,           /* handle to Store object            */
    Pint           *err_ind,        /* OUT error indicator               */
    Pint           *max_buf_size,   /* OUT max. input buffer size        */
    Pint_list      **pet_list,      /* OUT list of prompt and echo types */
    Plimit         *echo_area,      /* OUT default echo area             */
    Pstring_data   **string_data    /* OUT default data record           */
);

/* INQUIRE SET OF WORKSTATIONS TO WHICH POSTED */
extern void pinq_wss_posted(
    Pint        struct_id,            /* structure identifier      */
    Pint        num_elems_appl_list,  /* # elems in appl list      */
    Pint        start_ind,            /* starting index            */
    Pint        *err_ind,             /* OUT error indicator       */
    Pint_list   *ws,                  /* OUT list of workstations  */
    Pint        *num_elems_impl_list  /* OUT # elems in impl list  */
);

/* INQUIRE OPEN STRUCTURE */
extern void pinq_open_struct(
    Pint                  *err_ind,   /* OUT error indicator          */
    Popen_struct_status   *status,    /* OUT status of open structure */
    Pint                  *struct_id  /* OUT structure identifier     */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INQUIRE ELEMENT POINTER */
extern void pinq_elem_ptr(
    Pint   *err_ind,         /* OUT error indicator         */
    Pint   *elem_ptr_value   /* OUT element pointer value   */
);

/* INQUIRE CURRENT ELEMENT TYPE AND SIZE */
extern void pinq_cur_elem_type_size(
    Pint         *err_ind,     /* OUT error indicator   */
    Pelem_type   *elem_type,   /* OUT element type      */
    size_t       *elem_size    /* OUT element size      */
);

/* INQUIRE CURRENT ELEMENT CONTENT */
extern void pinq_cur_elem_content(
    Pstore       store,        /* handle to Store object   */
    Pint         *err_ind,     /* OUT error indicator      */
    Pelem_data   **elem_data   /* OUT element data         */
);

/* INQUIRE ELEMENT TYPE AND SIZE */
extern void pinq_elem_type_size(
    Pint         struct_id,    /* structure identifier   */
    Pint         elem_num,     /* element number         */
    Pint         *err_ind,     /* OUT error indicator    */
    Pelem_type   *elem_type,   /* OUT element type       */
    size_t       *elem_size    /* OUT element size       */
);

/* INQUIRE ELEMENT CONTENT */
extern void pinq_elem_content(
    Pint         struct_id,    /* structure identifier     */
    Pint         elem_num,     /* element number           */
    Pstore       store,        /* handle to Store object   */
    Pint         *err_ind,     /* OUT error indicator      */
    Pelem_data   **elem_data   /* OUT data record          */
);

/* INQUIRE STRUCTURE STATUS */
extern void pinq_struct_status(
    Pint             struct_id,   /* structure identifier   */
    Pint             *err_ind,    /* OUT error indicator    */
    Pstruct_status   *status      /* OUT existence status   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

```
/* INQUIRE PATHS TO ANCESTORS */
extern void pinq_paths_ances(
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pint                *err_ind,    /* OUT error indicator      */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);

/* INQUIRE PATHS TO DESCENDANTS */
extern void pinq_paths_descs(
    Pint                struct_id,   /* structure identifier     */
    Ppath_order         order,       /* path order               */
    Pint                depth,       /* path depth               */
    Pstore              store,       /* handle to Store object   */
    Pint                *err_ind,    /* OUT error indicator      */
    Pelem_ref_list_list **paths      /* OUT structure path list  */
);

/* ELEMENT SEARCH */
extern void pelem_search(
    Pint                  struct_id,       /* structure identifier       */
    Pint                  start_elem,      /* starting element position  */
    Psearch_dir           dir,             /* search direction           */
    const Pelem_type_list *incl,           /* element incl. list         */
    const Pelem_type_list *excl,           /* element excl. list         */
    Pint                  *err_ind,        /* OUT error indicator        */
    Psearch_status        *status,         /* OUT search status          */
    Pint                  *found_elem_ptr  /* OUT found element pointer  */
);

/* INCREMENTAL SPATIAL SEARCH 3 */
extern void pincr_spa_search3(
    const Ppoint3         *ref_pt,             /* search reference point     */
    Pfloat                dist,                /* search distance            */
    const Pelem_ref_list  *sp,                 /* starting path list         */
    Pclip_ind             clip_ind,            /* modeling clip indicator    */
    Pint                  ceiling,             /* search ceiling index       */
    const Pfilter_list    *norm,               /* normal filter list         */
    const Pfilter_list    *inv,                /* inverted filter list       */
    Pint                  num_elems_appl_list, /* # elems in appl list       */
    Pint                  start_ind,           /* starting index             */
    Pint                  *err_ind,            /* OUT error indicator        */
    Pelem_ref_list        *fp,                 /* OUT found path             */
    Pint                  *num_elems_impl_list /* OUT # elems in impl list   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
/* INCREMENTAL SPATIAL SEARCH */
extern void pincr_spa_search(
    const Ppoint          *ref_pt,            /* search reference point     */
    Pfloat                dist,               /* search distance            */
    const Pelem_ref_list  *sp,                /* starting path list         */
    Pclip_ind             clip_ind,           /* modeling clip indicator    */
    Pint                  ceiling,            /* search ceiling index       */
    const Pfilter_list    *norm,              /* normal filter list         */
    const Pfilter_list    *inv,               /* inverted filter list       */
    Pint                  num_elems_appl_list,/* # elems in appl list       */
    Pint                  start_ind,          /* starting index             */
    Pint                  *err_ind,           /* OUT error indicator        */
    Pelem_ref_list        *fp,                /* OUT found path             */
    Pint                  *num_elems_impl_list /* OUT # elems in impl list  */
);

/* INQUIRE INPUT QUEUE OVERFLOW */
extern void pinq_in_overf(
    Pint       *err_ind,    /* OUT error indicator          */
    Pint       *ws_id,      /* OUT workstation identifier   */
    Pin_class  *in_class,   /* OUT input class              */
    Pint       *in_num      /* OUT input device number      */
);

/* INQUIRE ERROR HANDLING MODE */
extern void pinq_err_hand_mode(
    Pint       *err_ind,   /* OUT error indicator   */
    Perr_mode  *err_mode   /* OUT error mode        */
);

/* EMERGENCY CLOSE PHIGS */
extern void pemergency_close_phigs(
        void
);

/* ERROR HANDLING */
extern void perr_hand(
    Pint          error_num,    /* error number                                  */
    Pint          func_num,     /* identifier of function that detected the error */
    const char    *error_file   /* name of error file                            */
);

/* ERROR LOGGING */
extern void perr_log(
    Pint          error_num,    /* error number                                  */
    Pint          func_num,     /* identifier of function that detected the error */
    const char    *error_file   /* name of error file                            */
);

/* SET ERROR HANDLING MODE */
extern void pset_err_hand_mode(
    Perr_mode  error_mode   /* error handling mode   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577581 330 ■


```
/* ESCAPE */
extern void pescape(
    Pint                    func_id,     /* escape function identifier       */
    const Pescape_in_data   *in_data,    /* input data for the function      */
    Pstore                  store,       /* handle to Store object           */
    Pescape_out_data        **out_data   /* OUT output data of the function  */
);

/* SET ERROR HANDLING */
extern void pset_err_hand(
    void   (*new_err_hand)(),   /* application's error handling function  */
    void   (**old_err_hand)()   /* OUT old error handling function        */
);

/* CREATE STORE */
extern void pcreate_store(
    Pint     *err_ind,   /* OUT error indicator   */
    Pstore   *store      /* OUT new Store         */
);

/* DELETE STORE */
extern void pdel_store(
    Pint     *err_ind,   /* OUT error indicator          */
    Pstore   *store      /* IN/OUT Store to be deleted   */
);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex B

(informative)

Example Programs

Annex B lists example programs.

B.1 star

```
/*
 * Program star
 *
 * Description:
 *   This program draws a yellow star on a blue background and writes
 *   the title 'Star' in white under the star.
 *
 * Define PHIGS constants.
 *
 */

#include <stdio.h>
#include "phigs.h"

/* define the coordinates of the star */
static  Ppoint  star[5] = {
    { 0.951057,  0.309017},
    {-0.951057,  0.309017},
    { 0.587785, -0.951057},
    { 0.0,       1.0},
    {-0.587785, -0.951057}
};

/* identity matrix used in the view representation */
static  Pmatrix identity = {
    { 1.0, 0.0, 0.0},
    { 0.0, 1.0, 0.0},
    { 0.0, 0.0, 1.0}
};

const Pint      ws_id = 1;                      /* id for workstation */

/* these two constants should be set to implementation specific values */
const char      *ws_conn_id = (void*)"";        /* connection id  */
const Pint      ws_type = 1;                    /* workstation type */

const Pint      struct_id = 1;


main(void)
{
    Pview_map           mapping;        /* view mapping information */
    Pint                error_ind;      /* error indicator */
    Pview_rep           view;           /* view representation */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
Pcolr_rep           colr;           /* a colour */
Ptext_align         align;          /* text alignment */
Ppoint              text_pt;        /* text point */
Ppoint_list         pl;             /* list of points for the star */


/* open PHIGS and a workstation. */
popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);

popen_ws(ws_id, ws_conn_id, ws_type);

/* Center the window around the origin. */
mapping.win.x_min = -1.25;
mapping.win.x_max =  1.25;
mapping.win.y_min = -1.25;
mapping.win.y_max =  1.25;

mapping.proj_vp.x_min = 0.0;
mapping.proj_vp.x_max = 1.0;
mapping.proj_vp.y_min = 0.0;
mapping.proj_vp.y_max = 1.0;
peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);

bcopy((char *)identity,
      (char *)view.ori_matrix,
      sizeof(Pmatrix));

view.clip_limit = mapping.proj_vp;
view.xy_clip = PIND_CLIP;
pset_view_rep(ws_id, 1, &view);

/* background is blue */
colr.rgb.red   = 0.0;
colr.rgb.green = 0.0;
colr.rgb.blue  = 1.0;
pset_colr_rep(ws_id, 0, &colr);

/* colour 1 is yellow */
colr.rgb.red   = 1.0;
colr.rgb.green = 1.0;
colr.rgb.blue  = 0.0;
pset_colr_rep(ws_id, 1, &colr);

/* colour 2 is white */
colr.rgb.red   = 1.0;
colr.rgb.green = 1.0;
colr.rgb.blue  = 1.0;
pset_colr_rep(ws_id, 2, &colr);


/* Post the structure to the workstation. */
ppost_struct(ws_id, struct_id, 1.0);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    /* Open the structure. */
    popen_struct(struct_id);

    /* Use the view we have established */
    pset_view_ind(1);

    /* Fill the star with solid yellow. */
    pset_int_style(PSTYLE_SOLID);
    pset_int_colr_ind(1);

    /* Draw the star. */
    pl.num_points = 5;
    pl.points = star;
    pfill_area(&pl);

    /* Select large characters centered under the star. */
    pset_char_ht(0.15);
    align.hor = PHOR_CTR;
    align.vert = PVERT_HALF;
    pset_text_align(&align);
    pset_text_colr_ind(2);

    /* Draw the title. */
    text_pt.x =  0.0;
    text_pt.y = -1.0;
    ptext(&text_pt, "Star");

    /* admire the star */
    sleep(10);

    /* Close the structure, unpost it and close the workstation and PHIGS. */
    pclose_struct();
    punpost_struct(ws_id, struct_id);
    pclose_ws(ws_id);
    pclose_phigs();
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

B.2 iron

```
/*
 * Program iron
 *
 * Description:
 *   This program draws a horizontal bar chart illustrating costs within
 *   the iron industry. The user can select the data to be displayed using
 *   a PHIGS choice device. The plot is adapted from 'Scientific American'
 *   May 1984 page 39.
 * Conformance:
 *   Choice device must support prompt and echo type 3
 *
 */

#include <stdio.h>
#include "phigs.h"


const Pint cho_dev              = 1;    /* choice device number */
const Pint str_pet              = 3;    /* choice prompt & echo type */

const Pint green                = 1;    /* colour index mapped to green */
const Pint red                  = 2;    /* colour index mapped to red */

/* the datasets to display */
#define NUM_COUNTRIES           3
#define NUM_DATA                6
#define NUM_PROMPTS             (NUM_COUNTRIES + 1)

/* constants used to reference structures */
const Pint struct_us            = 1;
const Pint struct_germany       = 2;
const Pint struct_japan         = 3;
const Pint struct_border        = 4;


const Pint      ws_id = 1;                      /* id for workstation */

/* these two constants should be set to implementation specific values */
const void      *ws_conn_id = (void*)"";        /* connection id  */
const Pint      ws_type = 1;                    /* workstation type */



typedef struct {
    char        *name;
    Pfloat      data[2][NUM_DATA];
} Dataset;

Dataset database[] = {
    {   "U.S.",
        {
            {   69.0, 50.0, 15.0, 53.0, 57.0, 150.0},
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            {   72.0, 50.0, 103.0, 0.0, 0.0, 56.0}
        }
    },
    {   "West Germany",
        {
            {   65.0, 42.0, 3.0, 89.0, 52.0, 93.0},
            {   70.0, 53.0, 102.0, 0.0, 0.0, 49.0}
        }
    },
    {   "Japan",
        {
            {   65.0, 47.0, 2.0, 60.0, 52.0, 55.0},
            {   70.0, 57.0, 105.0, 0.0, 0.0, 41.0},
        }
    }
};


char *labels[NUM_DATA] = {
    "Labour",
    "Iron Ore",
    "Coke or Coal",
    "Purchased Scrap",
    "Other Costs",
    "Other Energy"
};

/* internal routines */

static  void    init_phigs(Pint ws_id);
static  void    build_structures(void);
static  void    border(void);
static  void    draw(Dataset *nation);
static  void    ticks(Pfloat start, Pfloat end, Ptext_align *align);
static  void    bar(Pfloat length, Pfloat position);


main(void)
{
    Pint                current_country;
    Pin_status          status;
    Pint                selection;


    init_phigs(ws_id);
    build_structures();

    ppost_struct(ws_id, struct_border, 1.0);
    ppost_struct(ws_id, struct_us, 0.5);
    current_country = struct_us;

    for (;;) {
        preq_choice(ws_id, cho_dev, &status, &selection);
        if (status != PIN_STATUS_OK)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            break;                                  /* choice failed */
        if (selection > NUM_COUNTRIES)
            break;                                  /* user selected Exit */
        punpost_struct(ws_id, current_country);
        ppost_struct(ws_id, (current_country = selection), 0.5);
    }

    pclose_ws(ws_id);
    pclose_phigs();
}


static void
init_phigs(Pint ws_id)
{
    Pview_map           mapping;        /* view information */
    Pview_rep           view;           /* view representation */
    Pcolr_rep           colr;           /* colour representation */
    Pop_mode            mode;           /* device operating mode */
    Pecho_switch        esw;            /* echo switch */
    Pin_status          init_status;    /* initial status */
    Pint                init_choice;    /* initial choice */
    Pint                pet;            /* prompt/echo type */
    Plimit              echo_area;      /* echo area */
    Pchoice_data        *data_rec;      /* choice data record */
    char                *prompts[(NUM_COUNTRIES+1)]; /* prompts for choice */
    int                 i;              /* counter */
    Pstore              store;          /* store for choice data memory */
    Pint                error_ind;      /* error indicator */

    static      Pmatrix identity = {
        {   1.0, 0.0, 0.0},
        {   0.0, 1.0, 0.0},
        {   0.0, 0.0, 1.0}
    };


    popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);
    popen_ws(ws_id, ws_conn_id, ws_type);

    /*
     * define view
     */
    mapping.win.x_min = -115.0;
    mapping.win.x_max =  160.0;
    mapping.win.y_min =   -2.0;
    mapping.win.y_max =   14.0;

    mapping.proj_vp.x_min = 0.0;
    mapping.proj_vp.x_max = 1.0;
    mapping.proj_vp.y_min = 0.0;
    mapping.proj_vp.y_max = 1.0;
    peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    bcopy((char *)identity,
          (char *)view.ori_matrix,
          sizeof(Pmatrix));

    view.clip_limit = mapping.proj_vp;
    view.xy_clip = PIND_CLIP;
    pset_view_rep(ws_id, 1, &view);

    /*
     * define colours
     */
    colr.rgb.red   = 0.0;
    colr.rgb.green = 1.0;
    colr.rgb.blue  = 0.0;
    pset_colr_rep(ws_id, green, &colr);

    colr.rgb.red   = 1.0;
    colr.rgb.green = 0.0;
    colr.rgb.blue  = 0.0;
    pset_colr_rep(ws_id, red, &colr);


    /*
     * initial input device
     */
    pcreate_store(&error_ind, &store);

    pinq_choice_st(ws_id, cho_dev, store, &error_ind,
                   &mode, &esw, &init_status, &init_choice,
                   &pet, &echo_area, &data_rec);

    if (error_ind != PE_NO_ERROR) {
        fprintf(stderr, "Inquiry of choice device state failed (%d).\n",
                error_ind);
        exit(2);
    }

    for (i = 0; i < NUM_COUNTRIES; i++)
        prompts[i] = database[i].name;
    prompts[NUM_COUNTRIES] = "Exit";

    data_rec->pets.pet_r3.strings = prompts;
    data_rec->pets.pet_r3.num_strings = NUM_PROMPTS;

    pinit_choice(ws_id, cho_dev,
                 init_status,
                 init_choice,
                 str_pet,
                 &echo_area,
                 data_rec);

    pdel_store(&error_ind, &store);
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
static void
build_structures(void)
{
    int         i;

    popen_struct(struct_border);
    border();
    pclose_struct();

    for (i = struct_us; i <= struct_japan; i++) {
        popen_struct(i);
        draw(&database[i-1]);
        pclose_struct();
    }
}


static void
border(void)
{
    static      Ppoint box[5] = {
        {  0.0,  0.0},
        {150.0,  0.0},
        {150.0, 12.0},
        {  0.0, 12.0},
        {  0.0,  0.0}
    };
    static      Ptext_align     cen_bot = {PHOR_CTR, PVERT_BOTTOM};
    static      Ptext_align     cen_top = {PHOR_CTR, PVERT_TOP};
    static      Ptext_align     left_half = {PHOR_LEFT, PVERT_HALF};
    static      Ppoint          pos = {-114.0, 0.0};
    static      Ppoint          prod_pos = {0.0, -2.0};
    int         i;
    Ppoint_list pl;

    pl.num_points = 5;
    pl.points = box;
    ppolyline(&pl);

    pset_text_align(&left_half);
    pset_char_ht(0.5);
    pset_text_colr_ind(green);
    for (i = 0; i < NUM_DATA; i++) {
        pos.y = 1.2 + i * 2.0;
        ptext(&pos, labels[i]);
    }

    ticks(12.0, 12.2, &cen_bot);
    pset_text_colr_ind(red);
    ticks(0.0, -0.2, &cen_top);

    pset_text_colr_ind(green);
    ptext(&prod_pos, "Production Cost");
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
static void
draw(Dataset *nation)
{
    Pint        bar_data, colour;
    Pfloat      offset;
    static      Ppoint          nation_pos = {0.0, 14.0};
    static      Ptext_align     cen_top = {PHOR_CTR, PVERT_TOP};

    pset_text_colr_ind(green);
    pset_char_ht(0.7);
    pset_text_align(&cen_top);
    ptext(&nation_pos, nation->name);

    for (colour = green, offset = 1.6; colour <= red; colour++,offset -= 0.8) {
        pset_int_colr_ind(colour);
        pset_text_colr_ind(colour);
        pset_int_style(PSTYLE_SOLID);
        for (bar_data = 0; bar_data < NUM_DATA; bar_data++) {
            bar(nation->data[colour][bar_data], (2.0 * offset));
        }
    }
}



static void
ticks(Pfloat            tstart,
      Pfloat            tend,
      Ptext_align       *alignment)
{
#define         NUM_TICKS       4
    char        label[5];
    Ppoint      xy[2];
    Ppoint_list pl;
    int         i;

    xy[0].y = tstart;
    xy[1].y = tend;
    pl.num_points = 2;
    pl.points = xy;

    pset_text_align(alignment);

    for (i = 0; i < NUM_TICKS; i++) {
        xy[0].x = xy[1].x = 50.0 * i;
        ppolyline(&pl);
        sprintf(label, "%d", (int)xy[0].x);
        ptext(&xy[1], label);
    }
}

static void
bar(Pfloat      length,
    Pfloat      pos)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

Annex B

```
{
    static      Ptext_align     alignment = {PHOR_LEFT, PVERT_HALF};
    Ppoint      xy[4];
    Ppoint_list pl;

    pl.num_points = 4;
    pl.points = xy;

    if (length <= 0.0) {
        pset_text_align(&alignment);
        xy[0].x = 0.0;
        xy[0].y = pos;
        ptext(&xy[0], "0");
    }
    else {
        xy[0].x = xy[3].x = 0.0;
        xy[1].x = xy[2].x = length;

        xy[0].y = xy[3].y = pos + 0.4;
        xy[1].y = xy[2].y = pos - 0.4;
        pfill_area(&pl);
    }
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

B.3 dyna_star

```
/*
 * Program dyna_star
 *
 * Description:
 *   This program draws a yellow star on a blue background and writes
 *   the title 'Star' in white under the star.
 *   It changes the viewing parameters to zoom and pan on the star
 *   independent of the title.
 *   It then demonstrates some of the dynamics of PHIGS by rolling
 *   the star back and forth across the display by modifying the
 *   star's modelling transformation.
 *
 */

#include <stdio.h>
#include "phigs.h"

/* define the coordinates of the star */
static  Ppoint  star[5] = {
    { 0.951057,  0.309017},
    {-0.951057,  0.309017},
    { 0.587785, -0.951057},
    { 0.0,       1.0},
    {-0.587785, -0.951057}
};

/* identity matrix used in the view representation */
static  Pmatrix identity = {
    { 1.0, 0.0, 0.0},
    { 0.0, 1.0, 0.0},
    { 0.0, 0.0, 1.0}
};

const Pint      ws_id = 1;                      /* id for workstation */

/* these two constants should be set to implementation specific values */
const void      *ws_conn_id = (void *)"";       /* connection id  */
const Pint      ws_type = 1;                    /* workstation type */

const Pint      struct_id = 1;


main(void)
{
    Pview_map           mapping;        /* view information */
    Pint                error_ind;      /* error indicator */
    Pview_rep           view;           /* view representation */
    Pcolr_rep           colr;           /* a colour */
    Ptext_align         align;          /* text alignment */
    Ppoint              text_pt;        /* text point */
    int                 i;              /* counter */
    Pfloat              zoom;           /* zoom factor */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
Pmatrix             xform;          /* transformation matrix */
Pvec                scale_vec;      /* scale vector */
Pvec                shift_vec;      /* shift vector */
Ppoint              xform_pt;       /* point to transform about */
Ppoint_list         pl;             /* list of points for the star */

/* open PHIGS and a workstation. */
popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);

popen_ws(ws_id, ws_conn_id, ws_type);

/*
 * Define two 2D viewing transformations centered around the origin.
 * One will be used for the title and will remained unchanged.  The
 * other will be used for the star and will be changed for panning and
 * zooming.
 */
 /* Center the window around the origin. */
mapping.win.x_min = -1.25;
mapping.win.x_max =  1.25;
mapping.win.y_min = -1.25;
mapping.win.y_max =  1.25;

mapping.proj_vp.x_min = 0.0;
mapping.proj_vp.x_max = 1.0;
mapping.proj_vp.y_min = 0.0;
mapping.proj_vp.y_max = 1.0;
peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);

bcopy((char *)identity,
      (char *)view.ori_matrix,
      sizeof(Pmatrix));

view.clip_limit = mapping.proj_vp;
view.xy_clip = PIND_CLIP;
pset_view_rep(ws_id, 1, &view);
pset_view_rep(ws_id, 2, &view);


/* background is blue */
colr.rgb.red   = 0.0;
colr.rgb.green = 0.0;
colr.rgb.blue  = 1.0;
pset_colr_rep(ws_id, 0, &colr);

/* colour 1 is yellow */
colr.rgb.red   = 1.0;
colr.rgb.green = 1.0;
colr.rgb.blue  = 0.0;
pset_colr_rep(ws_id, 1, &colr);

/* colour 2 is white */
colr.rgb.red   = 1.0;
colr.rgb.green = 1.0;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    colr.rgb.blue  = 1.0;
    pset_colr_rep(ws_id, 2, &colr);


    /* Open the structure. */
    popen_struct(struct_id);

    /* Use the view we have established for the star */
    pset_view_ind(1);

    /* Fill the star with solid yellow. */
    pset_int_style(PSTYLE_SOLID);
    pset_int_colr_ind(1);

    /* Draw the star. */
    pl.num_points = 5;
    pl.points = star;
    pfill_area(&pl);

    /* Use the view for the title */
    pset_view_ind(2);

    /* Select large characters centered under the star. */
    pset_char_ht(0.15);
    align.hor = PHOR_CTR;
    align.vert = PVERT_HALF;
    pset_text_align(&align);
    pset_text_colr_ind(2);

    /* Draw the title. */
    text_pt.x =  0.0;
    text_pt.y = -1.0;
    ptext(&text_pt, "Star");

    /* close the structure */
    pclose_struct();

   /*
    * Set the deferral and update modes such that changes happen immediately.
    */
    pset_disp_upd_st(ws_id, PDEFER_ASAP, PMODE_UWOR);

   /*
    * Post the structure to the open workstation.
    */
    ppost_struct(ws_id, struct_id, 1.0);

    /*
     * Successively change the view mapping to zoom in and out.
     */
    for (i = 2; i <= 10; i++) {
        zoom = 1.25 / (Pfloat)i;
        mapping.win.x_min = -zoom;
        mapping.win.x_max =  zoom;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        mapping.win.y_min = -zoom;
        mapping.win.y_max =  zoom;
        peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);
        pset_view_rep(ws_id, 1, &view);
    }

    for (i = 9; i >= 1; i--) {
        zoom = 1.25 / (Pfloat)i;
        mapping.win.x_min = -zoom;
        mapping.win.x_max =  zoom;
        mapping.win.y_min = -zoom;
        mapping.win.y_max =  zoom;
        peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);
        pset_view_rep(ws_id, 1, &view);
    }

    /*
     * Successively change the view mapping. to pan back and forth across
     * the display.
     */
    for (i = 1; i <= 5; i++) {
        mapping.win.x_min = -1.25 - ((Pfloat)i * 0.1);
        mapping.win.x_max =  1.25 - ((Pfloat)i * 0.1);
        mapping.win.y_min = -1.25;
        mapping.win.y_max =  1.25;
        peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);
        pset_view_rep(ws_id, 1, &view);
    }

    for (i = 4; i >= 0; i--) {
        mapping.win.x_min = -1.25 - ((Pfloat)i * 0.1);
        mapping.win.x_max =  1.25 - ((Pfloat)i * 0.1);
        mapping.win.y_min = -1.25;
        mapping.win.y_max =  1.25;
        peval_view_map_matrix(&mapping, &error_ind, view.map_matrix);
        pset_view_rep(ws_id, 1, &view);
    }

    /*
     * Successively change the modelling transformation to "roll" the
     * Star back and forth across the display surface.
     *
     * Open up the structure so it can be edited.
     */
    popen_struct(1);

    /*
     * Insulate the title from changes to the modelling for the star.
     */
    pset_elem_ptr(4);
    pset_local_tran(identity, PTYPE_REPLACE);

    /*
     *  First scale it down to get something more "rollable".
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
 */
scale_vec.delta_x = 0.5;
scale_vec.delta_y = 0.5;
pscale(&scale_vec, &error_ind, xform);

pset_elem_ptr(3);
pset_local_tran(xform, PTYPE_REPLACE);

/*
 * Insert the rolling transformation as identity for now.
 */
pset_local_tran(identity, PTYPE_POSTCONCAT);

/*
 * Set up deferral mode so we don't get the deletes in it.
 */
pset_disp_upd_st(ws_id, PDEFER_WAIT, PMODE_NIVE);

/*
 * All of the following modifications take place on the same element
 * of the open structure.  Set the element pointer there and use replace
 * edit mode.
 */

pset_elem_ptr(5);

pset_edit_mode(PEDIT_REPLACE);

xform_pt.x = 0.0;
xform_pt.y = 0.0;

shift_vec.delta_y = 0.0;

scale_vec.delta_x = 1.0;
scale_vec.delta_y = 1.0;

for (i = 1; i <= 10; i++) {
    shift_vec.delta_x = -(i*0.05);
    pbuild_tran_matrix(&xform_pt, &shift_vec, -(0.5236*i),
                       &scale_vec, &error_ind, xform);
    pset_local_tran(xform, PTYPE_POSTCONCAT);
    pupd_ws(ws_id, PFLAG_PERFORM);
}
/* roll back to the right */
for (i = 9; i >= 0; i--) {
    shift_vec.delta_x = -(i*0.05);
    pbuild_tran_matrix(&xform_pt, &shift_vec, (0.5236*i),
                       &scale_vec, &error_ind, xform);
    pset_local_tran(xform, PTYPE_POSTCONCAT);
    pupd_ws(ws_id, PFLAG_PERFORM);
}

/* Let's roll all the way to the right. */
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577597 7T8 ■


```
    for (i = 1; i <= 10; i++) {
        shift_vec.delta_x = (i*0.05);
        pbuild_tran_matrix(&xform_pt, &shift_vec, (0.5236*i),
                           &scale_vec, &error_ind, xform);
        pset_local_tran(xform, PTYPE_POSTCONCAT);
        pupd_ws(ws_id, PFLAG_PERFORM);
    }
    /* roll back to the left,  back to where we started */
    for (i = 9; i >= 0; i--) {
        shift_vec.delta_x = (i*0.05);
        pbuild_tran_matrix(&xform_pt, &shift_vec, -(0.5236*i),
                           &scale_vec, &error_ind, xform);
        pset_local_tran(xform, PTYPE_POSTCONCAT);
        pupd_ws(ws_id, PFLAG_PERFORM);
    }

    /* Close the structure, close the workstation and PHIGS. */
    pclose_struct();
    pclose_ws(ws_id);
    pclose_phigs();
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

B.4 show_line

```
/*
 * Program: show_line
 *
 * Description:
 *   This program shows the available linetypes on a
 *   user selected workstation. It contains PHIGS initialization and
 *   shutdown modules and demonstrates how to program modules
 *   which do not change any statelist entries.
 *
 *
 */


#include <stdio.h>
#include "phigs.h"

static void     init_phigs(Pint *ws_type,  Pint *err_ind);
static void     line_demo(Pint ws_type,  Pint *err_ind);
static void     shutdown_phigs(void);


#ifndef TRUE
#define TRUE (1)
#endif

#ifndef FALSE
#define FALSE (!TRUE)
#endif

const Pint      ws_id     = 1;  /* id for workstation */
const Pint      struct_id = 1;  /* structure id */

main(void)
{
    Pint        ws_type;
    Pint        err_ind;

    /* call initialization module */
    init_phigs(&ws_type, &err_ind);

    /* Call demonstration module for linetype capabilities. */
    line_demo(ws_type, &err_ind);

    /* shutdown everything */
    shutdown_phigs();
}


static void
init_phigs(Pint         *ws_type,
           Pint         *err_ind)
{
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define NUM_TYPES  (10)          /* chunk size */
    Pint_list   type_list;              /* list of ws types structure */
    Pint        types[NUM_TYPES];       /* space for list of types */
    Pint        start_pos;              /* start position in returned list */
    Pint        length_list;            /* total length of the list */
    char        conn_id[80];            /* connection id from user */
    Psys_st     sys_st;                 /* system state */
    Pws_st      ws_st;                  /* workstation state */
    int         done;                   /* loop control */
    int         i;

    popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);

    /* Inquire available workstation types and print them */

    /* Access the list of available ws. types in "chunks" of
     * NUM_TYPES.
     */

    printf("The available OUTPUT and OUTIN workstation types are:\n");

    start_pos = 0;
    done = FALSE;
    while (!done) {
        pinq_list_avail_ws_types(NUM_TYPES, start_pos, err_ind,
                                 &type_list, &length_list);
        if (*err_ind == PE_NO_ERROR) {
            for (i = 0; i < type_list.num_ints; i++) {
                Pws_cat cat;            /* workstation category */

                pinq_ws_cat(type_list.ints[i], err_ind, &cat);
                if ((cat == PCAT_OUT) || (cat == PCAT_OUTIN))
                    printf("ws type: %d\n", type_list.ints[i]);
            }
            start_pos += NUM_TYPES;
        }
        else
            done = TRUE;
    }

    /* choose one workstation and open it. */

    printf("Please enter a workstation type (an integer)");
    printf("and a connection identifier (a string)\n");
    scanf("%d %s", ws_type, conn_id);

    popen_ws(ws_id, (void*)conn_id, *ws_type);

    /* Check operating state to ensure successful opening and activating. */
    pinq_sys_st(&sys_st);

    if (sys_st != PSYS_ST_PHOP)
        *err_ind = 3;
    else {
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        pinq_ws_st(&ws_st);
        if (ws_st != PWS_ST_WSOP)
            *err_ind = 3;
        else
            *err_ind = 0;
    }

}


static void
shutdown_phigs(void)
{
    pclose_ws(ws_id);
    pclose_phigs();
}



/* Demonstrate all available linetypes.
 * This module assumes one workstation with workstation identifier 1
 * and workstation type WTYPE to be open.
 * The display surface is not cleared and the OUTPUT stretches
 * across the current workstation viewport.
 */
static void
line_demo(Pint          ws_type,
          Pint          *err_ind)
{
    Psys_st     sys_st;                 /* system state */
    Pws_st      ws_st;                  /* workstation state */
    Pupd_st     upd_st;                 /* update state */
    Plimit      req_win;                /* requested ws window */
    Plimit      req_vp;                 /* requested ws viewport  */
    Plimit      cur_win;                /* current ws window */
    Plimit      cur_vp;                 /* current ws viewport  */
    Pint        tot_length;             /* total length of list of line facilities */
    Pline_facs  ln_facs;                /* line facilities */
    char        buff[80];               /* buffer for number of line type */
    Pfloat      win_height;             /* window height */
    Pfloat      dist;                   /* distance */
    Pvec        char_up;                /* text attributes */
    Ptext_align align;
    Ppoint      pts[2];
    Ppoint_list pl;
    int         i;



    if (sys_st != PSYS_ST_PHOP)
        *err_ind = 3;
    else {
        pinq_ws_st(&ws_st);
        if (ws_st != PWS_ST_WSOP)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            *err_ind = 3;
        else
            *err_ind = 0;
    }
    if (*err_ind != 0)
        return;

    /* inquire workstation transform */
    pinq_ws_tran(ws_id, err_ind, &upd_st, &req_win, &cur_win, &req_vp, &cur_vp);

    /* inquire polyline line facilities. Indicate we have a zero-length
     * list and PHIGS will tell us how long it really is.
     */
    pinq_line_facs(ws_type, 0, 0, err_ind, &ln_facs, &tot_length);
    ln_facs.types.ints = (Pint *)malloc(sizeof(Pint) * tot_length);
    pinq_line_facs(ws_type, tot_length, 0, err_ind, &ln_facs, &tot_length);

    ppost_struct(struct_id, 1, 1.0);
    popen_struct(struct_id);

    /* Use individual attributes */
    pset_indiv_asf(PASPECT_LINETYPE, PASF_INDIV);
    pset_indiv_asf(PASPECT_LINEWIDTH, PASF_INDIV);
    pset_indiv_asf(PASPECT_LINE_COLR_IND , PASF_INDIV);

    pset_indiv_asf(PASPECT_TEXT_FONT, PASF_INDIV);
    pset_indiv_asf(PASPECT_TEXT_PREC, PASF_INDIV);
    pset_indiv_asf(PASPECT_CHAR_EXPAN, PASF_INDIV);
    pset_indiv_asf(PASPECT_CHAR_SPACE, PASF_INDIV);
    pset_indiv_asf(PASPECT_TEXT_COLR_IND, PASF_INDIV);

    pset_linewidth(1.0);
    pset_line_colr_ind(1);

    char_up.delta_x = 0.0;
    char_up.delta_y= 1.0;
    pset_char_up_vec(&char_up);

    pset_text_path(PPATH_RIGHT);

    align.hor = PHOR_LEFT;
    align.vert = PVERT_HALF;
    pset_text_align(&align);

    pset_text_font(1);
    pset_text_prec(PPREC_CHAR);
    pset_char_expan(1.0);
    pset_char_space(0.0);
    pset_text_colr_ind(1);

    /* Compute distance between lines */
    win_height = cur_win.y_max - cur_win.y_min ;
    dist = win_height / ln_facs.types.num_ints;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    /* Set character height to half of the distance between lines,
     * but not more than 1/20th of the height of the current
     * workstation viewport.
     */
    if (dist/2.0 < win_height/20.0)
        pset_char_ht(win_height/20.0);
    else
        pset_char_ht(dist/2.0);

    /* Lines stretch from the left bound to the middle of the
     * current workstation viewport
     */

    pts[0].x = cur_win.x_min;
    pts[1].x = (cur_win.x_min + cur_win.x_max) / 2.0;
    pts[0].y = cur_win.y_max - (dist / 2.0);
    pl.num_points = 2;
    pl.points = pts;



    /* loop over available line types */
    for (i = 0; i < ln_facs.types.num_ints; i++) {
        pset_linetype(ln_facs.types.ints[i]);
        pts[1].y = pts[0].y;
        ppolyline(&pl);
        pts[0].y -= dist;

        /* annotate linetypes */

        sprintf(buff, "%d", ln_facs.types.ints[i]);
        ptext(&pts[1], buff);
    }

    pclose_struct();
    err_ind = 0;
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

B.5 xform_pline

```
/*
 * Program xform_pline
 *
 * Description:
 *   This program applies one or more transforms to a closed
 *   polyline connection a user-supplied set of points and then
 *   draws the result on a plotter.
 *
 * Conformance:
 *   This example assumes that all the required input devices
 *   are available on the same OUTIN workstation.
 *
 */


#include <stdio.h>
#include "phigs.h"

typedef enum {
    POINTS_OK,
    POINTS_DONE
} Points_status;

/* Constants */

const Pint      display_ws_id = 1;      /* id for workstation to display
                                         * graphics.
                                         */
const Pint      display_stroke_num = 1; /* stroke device number */
const Pint      display_choice_num = 1; /* choice device number */
const Pint      display_val_num = 1;    /* valuator device number */

/* these two constants should be set to implementation specific values */
const char      *display_conn_id = (void*)"";   /* connection id  */
const Pint      display_ws_type = 1;            /* workstation type */



const Pint      plot_ws_id = 2;         /* id for workstation to plot
                                         * graphics.
                                         */
/* these two constants should be set to implementation specific values */
const char      *plot_conn_id = (void*)"";      /* connection id  */
const Pint      plot_ws_type = 2;               /* workstation type */

const Pint      line_struct_id = 1;
const Pint      text_struct_id = 2;

/* internal routines */
static  Pint            inq_stroke_buff_size(void);
static  Points_status   get_points(Ppoint_list *);
static  void            transform_points(Ppoint_list *pl);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
static  void            print_on_plotter(void);


main(void)
{
    Pview_map           mapping;        /* view mapping specification */
    Pview_rep           view;           /* view specification */
    Pint                err_ind;        /* error indicator  */
    Ppoint_list         pl;

    /* Perform initialization and opening of the OUTIN workstation. */
    popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);
    popen_ws(display_ws_id, display_conn_id, display_ws_type);

    /* allocate point list for stroke  */
    pl.num_points = inq_stroke_buff_size() + 1;
    pl.points = (Ppoint *)malloc(sizeof(Ppoint) * pl.num_points);


    /* Post the structure to the open workstation. */
    ppost_struct(line_struct_id,  1,  1.0);

    /*
     * Define two 2D viewing transformations centred around the origin.
     * One will be used for the title and will remained unchanged.  The
     * Other will be used for the star and will be changed for panning and
     * zooming.
     */

    mapping.win.x_min = -10.0;
    mapping.win.x_max =  10.0;
    mapping.win.y_min = -10.0;
    mapping.win.y_max =  10.0;

    mapping.proj_vp.x_min = 0.1;
    mapping.proj_vp.x_max = 0.9;
    mapping.proj_vp.y_min = 0.1;
    mapping.proj_vp.y_max = 0.9;

    peval_view_map_matrix(&mapping, &err_ind, view.map_matrix);

    view.ori_matrix[0][0] = 1.0;
    view.ori_matrix[0][1] = 0.0;
    view.ori_matrix[0][2] = 0.0;
    view.ori_matrix[1][0] = 0.0;
    view.ori_matrix[1][1] = 1.0;
    view.ori_matrix[1][2] = 0.0;
    view.ori_matrix[2][0] = 0.0;
    view.ori_matrix[2][1] = 0.0;
    view.ori_matrix[2][2] = 1.0;

    view.clip_limit = mapping.proj_vp;
    view.xy_clip = PIND_CLIP;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
pset_view_rep(display_ws_id, 1, &view);

/* Set the deferral and update modes so that changes happen immediately. */

pset_disp_upd_st(display_ws_id, PDEFER_ASAP,  PMODE_UWOR);

/* Open the structure. */
popen_struct(line_struct_id);

/*
 * Insert the local transformation as identity for now. Reuse
 * view.ori_matrix since that is an identity matrix
 */
pset_local_tran(view.ori_matrix, PTYPE_POSTCONCAT);

/* Use bundled attributes */
pset_indiv_asf(PASPECT_LINETYPE, PASF_BUNDLED);
pset_indiv_asf(PASPECT_LINEWIDTH, PASF_BUNDLED);
pset_indiv_asf(PASPECT_LINE_COLR_IND , PASF_BUNDLED);

pset_indiv_asf(PASPECT_TEXT_FONT, PASF_BUNDLED);
pset_indiv_asf(PASPECT_TEXT_PREC, PASF_BUNDLED);
pset_indiv_asf(PASPECT_CHAR_EXPAN, PASF_BUNDLED);
pset_indiv_asf(PASPECT_CHAR_SPACE, PASF_BUNDLED);
pset_indiv_asf(PASPECT_TEXT_COLR_IND, PASF_BUNDLED);

pset_line_colr_ind(3);

if (get_points(&pl) == POINTS_OK) {

    /* put points into structure as a closed polyline */

    /* make last pt the same as the 1st */
    pl.points[pl.num_points].x = pl.points[0].x;
    pl.points[pl.num_points].y = pl.points[0].y;
    pl.num_points++;
    ppolyline(&pl);

    /* Apply repeated transformations to the structure.
     * Set the element pointer on the local modelling transformation
     * and put the edit mode in REPLACE.
     */
    pset_elem_ptr(1);
    pset_edit_mode(PEDIT_REPLACE);

    transform_points(&pl);
}

pclose_struct();

print_on_plotter();

/* shut down PHIGS */
pclose_ws(display_ws_id);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    pclose_phigs();
}

static Pint
inq_stroke_buff_size(void)
{
    Pint                err;
    Pint                in_buf_size;
    Pint                max_buf_size;
    Plimit              echo_area;
    Pstore              store;
    Pint_list           *pets;
    Pint                max_num_pets;
    Pstroke_data        *stroke_data_ptr;

    pcreate_store(&err, &store);

    pinq_def_stroke_data(display_ws_type, display_stroke_num,
                         store, &err, &max_buf_size, &pets,
                         &echo_area, &stroke_data_ptr);
    if (err != PE_NO_ERROR) {
        fprintf(stderr, "inquiry failed for stroke data.\n");
        exit(1);
    }

    pdel_store(&err, &store);

    return(max_buf_size);
}


/*
 * Ask the user to enter a list of points via the stroke device.
 * Keep asking until she enters a valid number of points or we get
 * a bad status.
 */

static Points_status
get_points(Ppoint_list  *pl)
{
    Pin_status  status;
    Pint        view_ind;

    while (1) {
        printf("Stroke input needed for polyline ...\n");
        preq_stroke(display_ws_id, display_stroke_num,
                    &status, &view_ind, pl);

        printf("Got %d points\n", pl->num_points);

        if (status != PIN_STATUS_OK)
            return(POINTS_DONE);

        if ((view_ind == 1) && (pl->num_points >= 2))
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            return(POINTS_OK);
    }

}

static void
transform_points(Ppoint_list    *pl)
{
    Pin_status  status;
    Pint        choice;
    Pint        view_ind;
    Pmatrix     tran;
    Ppoint      fixed_point;
    Pvec        shift_vec;
    Pfloat      angle;
    Pvec        scale_vec;
    Pint        err_ind;
    Pfloat      val;

    tran[0][0] = 1.0;
    tran[0][1] = 0.0;
    tran[0][2] = 0.0;
    tran[1][0] = 0.0;
    tran[1][1] = 1.0;
    tran[1][2] = 0.0;
    tran[2][0] = 0.0;
    tran[2][1] = 0.0;
    tran[2][2] = 1.0;


    while (1) {
        printf("Choice inputed needed...\n");
        printf("\t1 - translate\n");
        printf("\t2 - rotate\n");
        printf("\t3 - scale\n");
        printf("\t4 - exit\n");

        preq_choice(display_ws_id, display_choice_num, &status, &choice);
        printf("Got choice %d\n", choice);

        if (status != PIN_STATUS_OK)
            return;

        switch (choice) {
          case 1:
            while (1) {
                printf("Stroke input needed. 2 pts for translate ...\n");
                preq_stroke(display_ws_id, display_stroke_num,
                            &status, &view_ind, pl);

                printf("Got %d points\n", pl->num_points);
                if (status != PIN_STATUS_OK)
                    return;
                if ((view_ind == 1) && pl->num_points == 2)
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            break;
        }
        fixed_point.x = 0.0;
        fixed_point.y = 0.0;
        shift_vec.delta_x = pl->points[1].x - pl->points[0].x;
        shift_vec.delta_y = pl->points[1].y - pl->points[0].y;
        angle = 0.0;
        scale_vec.delta_x = 1.0;
        scale_vec.delta_y = 1.0;
        pcompose_tran_matrix(tran, &fixed_point, &shift_vec,
                             angle, &scale_vec, &err_ind, tran);
        break;

    case 2:
        while (1) {
            printf("Valuator input needed for angle ...\n");
            preq_val(display_ws_id, display_val_num, &status, &angle);
            printf("Got angle = %f \n", angle);
            if (status == PIN_STATUS_OK)
                break;
        }
        fixed_point.x = 0.0;
        fixed_point.y = 0.0;
        shift_vec.delta_x = 0.0;
        shift_vec.delta_y = 0.0;
        scale_vec.delta_x = 1.0;
        scale_vec.delta_y = 1.0;
        pcompose_tran_matrix(tran, &fixed_point, &shift_vec,
                             angle, &scale_vec, &err_ind, tran);
        break;

    case 3:
        while (1) {
            printf("Valuator input needed X scale ...\n");
            preq_val(display_ws_id, display_val_num, &status, &val);
            scale_vec.delta_x = (Pfloat) val;
            scale_vec.delta_x *= 2.0;
            printf("Got X scale = %f \n", scale_vec.delta_x);
            if (status == PIN_STATUS_OK)
                break;
        }
        while (1) {
            printf("Valuator input needed Y scale ...\n");
            preq_val(display_ws_id, display_val_num, &status, &val);
            scale_vec.delta_y = val;
            scale_vec.delta_y *= 2.0;
            printf("Got Y scale = %f \n", scale_vec.delta_y);
            if (status == PIN_STATUS_OK)
                break;
        }
        fixed_point.x = 0.0;
        fixed_point.y = 0.0;
        shift_vec.delta_x = 0.0;
        shift_vec.delta_y = 0.0;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
            angle = 0.0;
            pcompose_tran_matrix(tran, &fixed_point, &shift_vec,
                                 angle, &scale_vec, &err_ind, tran);
            break;


          case 4:
            return;     /* finished with transforming */

          default:
            /* ignore choices other than 1 thru 4 */
            break;
        }

        pset_local_tran(tran, PTYPE_POSTCONCAT);
        pupd_ws(display_ws_id, PFLAG_PERFORM);
    } /* endwhile */
}

static void
print_on_plotter(void)
{
    Pcolr_rep           colr;
    Pline_bundle        line_bundle;
    Ptext_bundle        text_bundle;
    Ptext_align         align;
    Ppoint              pt;

    printf("Plotting on hardcopy device ...\n");

    popen_ws(plot_ws_id, plot_conn_id, plot_ws_type);

    colr.rgb.red   = 1.0;
    colr.rgb.green = 0.0;
    colr.rgb.blue  = 0.0;
    pset_colr_rep(plot_ws_id, 1, &colr);

    line_bundle.type = PLINE_SOLID;
    line_bundle.width = 1.5;
    line_bundle.colr_ind = 1;
    pset_line_rep(plot_ws_id, 3, &line_bundle);

    text_bundle.font  = 1;
    text_bundle.prec  = PPREC_STROKE;
    text_bundle.char_expan = 1.0;
    text_bundle.char_space = 0.0;
    text_bundle.colr_ind  = 1;
    pset_text_rep(plot_ws_id, 2, &text_bundle);

    /* Label the picture. */
    ppost_struct(plot_ws_id, text_struct_id, 1.0);
    popen_struct(text_struct_id);
    pset_char_ht(1.0);
    pset_text_ind(2);
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
    align.hor  = PHOR_NORM;
    align.vert = PVERT_BASE;
    pset_text_align(&align);

    pt.x = 0.0;
    pt.y = 0.0;
    ptext(&pt, "This is a polyline");

    pclose_struct();

    pupd_ws(plot_ws_id, PFLAG_PERFORM);

    /* Shut down ws */
    pclose_ws(plot_ws_id);
}
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577611 9T8 ■

Annex C

(informative)

Macros for short function identifiers

Annex C lists an example mapping of bound names to short identifiers.

C.1 Short function identifiers

Applications shall use the long function identifiers specified in clause 7. These macros define a mapping of long function identifiers to short identifiers for environments whose linker expects external identifiers to be less than 7 characters long. The short identifiers are the same as the bound names in ISO/IEC 9593-1 (PHIGS FORTRAN Binding).

```
#define    popen_phigs                popph
#define    pclose_phigs               pclph
#define    popen_ws                   popwk
#define    pclose_ws                  pclwk
#define    predraw_all_structs        prst
#define    pupd_ws                    puwk
#define    pset_disp_upd_st           psdus
#define    pmessage                   pmsg
#define    ppolyline3                 ppl3
#define    ppolyline                  ppl
#define    ppolymarker3               ppm3
#define    ppolymarker                ppm
#define    ptext3                     ptx3
#define    ptext                      ptx
#define    panno_text_rel3            patr3
#define    panno_text_rel             patr
#define    pfill_area3                pfa3
#define    pfill_area                 pfa
#define    pfill_area_set3            pfas3
#define    pfill_area_set             pfas
#define    pcell_array3               pca3
#define    pcell_array                pca
#define    pset_line_ind              pspli
#define    pset_marker_ind            pspmi
#define    pset_text_ind              pstxi
#define    pset_int_ind               psii
#define    pset_edge_ind              psedi
#define    pset_linetype              psln
#define    pset_linewidth             pslwsc
#define    pset_line_colr_ind         psplci
#define    pset_marker_type           psmk
#define    pset_marker_size           psmksc
#define    pset_marker_colr_ind       pspmci
#define    pset_text_font             pstxfn
#define    pset_text_prec             pstxpr
#define    pset_char_expan            pschxp
#define    pset_char_space            pschsp
#define    pset_text_colr_ind         pstxci
#define    pset_char_ht               pschh
#define    pset_char_up_vec           pschup
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     pset_text_path                  pstxp
#define     pset_text_align                 pstxal
#define     pset_anno_char_ht               psatch
#define     pset_anno_char_up_vec           psatcu
#define     pset_anno_path                  psatp
#define     pset_anno_align                 psatal
#define     pset_anno_style                 psans
#define     pset_int_style                  psis
#define     pset_int_style_ind              psisi
#define     pset_int_colr_ind               psici
#define     pset_edge_flag                  psedfg
#define     pset_edgetype                   psedt
#define     pset_edgewidth                  psewsc
#define     pset_edge_colr_ind              psedci
#define     pset_pat_size                   pspa
#define     pset_pat_ref_point_vecs         psprpv
#define     pset_pat_ref_point              psparf
#define     padd_names_set                  pads
#define     premove_names_set               pres
#define     pset_indiv_asf                  psiasf
#define     pset_line_rep                   psplr
#define     pset_marker_rep                 pspmr
#define     pset_text_rep                   pstxr
#define     pset_int_rep                    psir
#define     pset_edge_rep                   psedr
#define     pset_pat_rep                    pspar
#define     pset_colr_rep                   pscr
#define     pset_highl_filter               pshlft
#define     pset_invis_filter               psivft
#define     pset_colr_model                 pscmd
#define     pset_hlhsr_id                   pshrid
#define     pset_hlhsr_mode                 pshrm
#define     pset_local_tran3                pslmt3
#define     pset_local_tran                 pslmt
#define     pset_global_tran3               psgmt3
#define     pset_global_tran                psgmt
#define     pset_model_clip_vol3            psmcv3
#define     pset_model_clip_vol             psmcv
#define     pset_model_clip_ind             psmcli
#define     prestore_model_clip_vol         prmclv
#define     pset_view_ind                   psvwi
#define     pset_view_rep3                  psvwr3
#define     pset_view_rep                   psvwr
#define     pset_view_tran_in_pri           psvtip
#define     pset_ws_win3                    pswkw3
#define     pset_ws_win                     pswkw
#define     pset_ws_vp3                     pswkv3
#define     pset_ws_vp                      pswkv
#define     ptranslate3                     ptr3
#define     ptranslate                      ptr
#define     pscale3                         psc3
#define     pscale                          psc
#define     protate_x                       prox
#define     protate_y                       proy
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     protate_z                       proz
#define     protate                         pro
#define     pcompose_matrix3                pcom3
#define     pcompose_matrix                 pcom
#define     ptran_point3                    ptp3
#define     ptran_point                     ptp
#define     pbuild_tran_matrix3             pbltm3
#define     pbuild_tran_matrix              pbltm
#define     pcompose_tran_matrix3           pcotm3
#define     pcompose_tran_matrix            pcotm
#define     peval_view_ori_matrix3          pevom3
#define     peval_view_ori_matrix           pevom
#define     peval_view_map_matrix3          pevmm3
#define     peval_view_map_matrix           pevmm
#define     popen_struct                    popst
#define     pclose_struct                   pclst
#define     pexec_struct                    pexst
#define     plabel                          plb
#define     pappl_data                      pap
#define     pset_edit_mode                  psedm
#define     pcopy_all_elems_struct          pcelst
#define     pset_elem_ptr                   psep
#define     poffset_elem_ptr                posep
#define     pset_elem_ptr_label             pseplb
#define     pdel_elem                       pdel
#define     pdel_elem_range                 pdelra
#define     pdel_elems_labels               pdellb
#define     pempty_struct                   pemst
#define     pdel_struct                     pdst
#define     pdel_struct_net                 pdsn
#define     pdel_all_structs                pdas
#define     pchange_struct_id               pcstid
#define     pchange_struct_refs             pcstrf
#define     pchange_struct_id_refs          pcstir
#define     ppost_struct                    ppost
#define     punpost_struct                  pupost
#define     punpost_all_structs             pupast
#define     popen_ar_file                   poparf
#define     pclose_ar_file                  pclarf
#define     par_structs                     parst
#define     par_struct_nets                 parsn
#define     par_all_structs                 parast
#define     pset_conf_res                   pscnrs
#define     pret_struct_ids                 prsid
#define     pret_structs                    prest
#define     pret_struct_nets                presn
#define     pret_all_structs                prast
#define     pret_paths_ances                prepan
#define     pret_paths_descs                prepde
#define     pdel_structs_ar                 pdstar
#define     pdel_struct_nets_ar             pdsnar
#define     pdel_all_structs_ar             pdasar
#define     pset_pick_id                    pspkid
#define     pset_pick_filter                pspkft
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     pinit_loc3                  pinlc3
#define     pinit_loc                   pinlc
#define     pinit_stroke3               pinsk3
#define     pinit_stroke                pinsk
#define     pinit_val3                  pinvl3
#define     pinit_val                   pinvl
#define     pinit_choice3               pinch3
#define     pinit_choice                pinch
#define     pinit_pick3                 pinpk3
#define     pinit_pick                  pinpk
#define     pinit_string3               pinst3
#define     pinit_string                pinst
#define     pset_loc_mode               pslcm
#define     pset_stroke_mode            psskm
#define     pset_val_mode               psvlm
#define     pset_choice_mode            pschm
#define     pset_pick_mode              pspkm
#define     pset_string_mode            psstm
#define     preq_loc3                   prqlc3
#define     preq_loc                    prqlc
#define     preq_stroke3                prqsk3
#define     preq_stroke                 prqsk
#define     preq_val                    prqvl
#define     preq_choice                 prqch
#define     preq_pick                   prqpk
#define     preq_string                 prqst
#define     psample_loc3                psmlc3
#define     psample_loc                 psmlc
#define     psample_stroke3             psmsk3
#define     psample_stroke              psmsk
#define     psample_val                 psmvl
#define     psample_choice              psmch
#define     psample_pick                psmpk
#define     psample_string              psmst
#define     pawait_event                pwait
#define     pflush_events               pflush
#define     pget_loc3                   pgtlc3
#define     pget_loc                    pgtlc
#define     pget_stroke3                pgtsk3
#define     pget_stroke                 pgtsk
#define     pget_val                    pgtvl
#define     pget_choice                 pgtch
#define     pget_pick                   pgtpk
#define     pget_string                 pgtst
#define     pwrite_item                 pwitm
#define     pget_item_type              pgtitm
#define     pread_item                  prditm
#define     pinterpret_item             piitm
#define     pinq_sys_st                 pqsys
#define     pinq_ws_st                  pqwkst
#define     pinq_struct_st              pqstrs
#define     pinq_ar_st                  pqars
#define     pinq_list_avail_ws_types    pqewk
#define     pinq_phigs_facs             pqphf
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     pinq_gse_facs                   pqgsef
#define     pinq_model_clip_facs            pqmclf
#define     pinq_edit_mode                  pqedm
#define     pinq_open_wss                   pqopwk
#define     pinq_struct_ids                 pqsid
#define     pinq_ar_files                   pqarf
#define     pinq_conf_res                   pqcnrs
#define     pinq_all_conf_structs           pqcst
#define     pinq_conf_structs_net           pqcstn
#define     pinq_more_simult_events         pqsim
#define     pinq_ws_conn_type               pqwkc
#define     pinq_list_view_inds             pqevwi
#define     pinq_view_rep                   pqvwr
#define     pinq_hlhsr_mode                 pqhrm
#define     pinq_posted_structs             pqpost
#define     pinq_disp_upd_st                pqdus
#define     pinq_list_line_inds             pqepli
#define     pinq_line_rep                   pqplr
#define     pinq_list_marker_inds           pqepmi
#define     pinq_marker_rep                 pqpmr
#define     pinq_list_text_inds             pqetxi
#define     pinq_text_rep                   pqtxr
#define     pinq_list_int_inds              pqeii
#define     pinq_int_rep                    pqir
#define     pinq_list_edge_inds             pqeedi
#define     pinq_edge_rep                   pqedr
#define     pinq_list_pat_inds              pqepai
#define     pinq_pat_rep                    pqpar
#define     pinq_colr_model                 pqcmd
#define     pinq_list_colr_inds             pqeci
#define     pinq_colr_rep                   pqcr
#define     pinq_highl_filter               pqhlft
#define     pinq_invis_filter               pqivft
#define     pinq_ws_tran3                   pqwkt3
#define     pinq_ws_tran                    pqwkt
#define     pinq_loc_st3                    pqlcs3
#define     pinq_loc_st                     pqlcs
#define     pinq_stroke_st3                 pqsks3
#define     pinq_stroke_st                  pqsks
#define     pinq_val_st3                    pqvls3
#define     pinq_val_st                     pqvls
#define     pinq_choice_st3                 pqchs3
#define     pinq_choice_st                  pqchs
#define     pinq_pick_st3                   pqpks3
#define     pinq_pick_st                    pqpks
#define     pinq_string_st3                 pqsts3
#define     pinq_string_st                  pqsts
#define     pinq_ws_cat                     pqwkca
#define     pinq_disp_space_size3           pqdsp3
#define     pinq_disp_space_size            pqdsp
#define     pinq_hlhsr_id_facs              pqhrif
#define     pinq_hlhsr_mode_facs            pqhrmf
#define     pinq_view_facs                  pqvwf
#define     pinq_pred_view_rep              pqpvwr
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     pinq_ws_class                   pqwkcl
#define     pinq_dyns_ws_attrs              pqdswa
#define     pinq_def_disp_upd_st            pqddus
#define     pinq_line_facs                  pqplf
#define     pinq_pred_line_rep              pqpplr
#define     pinq_marker_facs                pqpmf
#define     pinq_pred_marker_rep            pqppmr
#define     pinq_text_facs                  pqtxf
#define     pinq_pred_text_rep              pqptxr
#define     pinq_anno_facs                  pqanf
#define     pinq_text_extent                pqtxx
#define     pinq_int_facs                   pqif
#define     pinq_pred_int_rep               pqpir
#define     pinq_edge_facs                  pqedf
#define     pinq_pred_edge_rep              pqpedr
#define     pinq_pat_facs                   pqpaf
#define     pinq_pred_pat_rep               pqppar
#define     pinq_colr_model_facs            pqcmdf
#define     pinq_colr_facs                  pqcf
#define     pinq_pred_colr_rep              pqpcr
#define     pinq_list_avail_gdp3            pqegd3
#define     pinq_list_avail_gdp             pqegdp
#define     pinq_gdp3                       pqgdp3
#define     pinq_gdp                        pqgdp
#define     pinq_list_avail_gse             pqegse
#define     pinq_num_disp_pris              pqdp
#define     pinq_ws_st_table                pqwksl
#define     pinq_dyns_structs               pqdstr
#define     pinq_num_avail_in               pqli
#define     pinq_def_loc_data3              pqdlc3
#define     pinq_def_loc_data               pqdlc
#define     pinq_def_stroke_data3           pqdsk3
#define     pinq_def_stroke_data            pqdsk
#define     pinq_def_val_data3              pqdvl3
#define     pinq_def_val_data               pqdvl
#define     pinq_def_choice_data3           pqdch3
#define     pinq_def_choice_data            pqdch
#define     pinq_def_pick_data3             pqdpk3
#define     pinq_def_pick_data              pqdpk
#define     pinq_def_string_data3           pqdst3
#define     pinq_def_string_data            pqdst
#define     pinq_wss_posted                 pqwkpo
#define     pinq_open_struct                pqopst
#define     pinq_elem_ptr                   pqep
#define     pinq_cur_elem_type_size         pqcets
#define     pinq_cur_elem_content           pqceco
#define     pinq_elem_type_size             pqets
#define     pinq_elem_content               pqeco
#define     pinq_struct_status              pqstst
#define     pinq_paths_ances                pqpan
#define     pinq_paths_descs                pqpde
#define     pelem_search                    pels
#define     pincr_spa_search3               piss3
#define     pincr_spa_search                piss
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
#define     pinq_in_overf                pqiqov
#define     pinq_err_hand_mode           pqerhm
#define     pemergency_close_phigs       peclph
#define     perr_hand                    perhnd
#define     perr_log                     perlog
#define     pset_err_hand_mode           pserhm
#define     pescape                      pesc
#define     pset_err_hand                pserh
#define     pcreate_store                pcrsto
#define     pdel_store                   pdlsto
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex D

(informative)

Memory management

D.1 Introduction

Several functions in this part of ISO/IEC 9593 return data whose length or size is not known *a priori* by the application. The memory management policies for these functions is described in 3.9. This annex discusses sample code and diagrams to help illustrate the policies.

Two memory management policies are defined in 3.9: The first is for functions which return a simple list of uniform elements; this policy is discussed in D.2. The other policy is for functions which return complex data structures and is discussed in D.3.

D.2 Functions that return simple lists

The auxiliary parameters `num_elems_appl_list`, `start_ind`, `num_elems_impl_list` are used by functions which return simple lists. These parameters allow the application to retrieve the contents of the implementation's list a piece at a time. The sample program in table D.1 illustrates the use of these parameters to retrieve a portion of the implementation's list.

The sample program inquires a piece of the list of polyline indices supported by a particular workstation by calling the function `pinq_list_line_inds()`. The program inquires a piece of the implementation's list that begins at index 2 and is 12 elements long.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table D.1 — Sample program of the memory management policy for simple lists

```
#include        "phigs.h"
main ()
{
    Pint_list  line_inds;
    Pint       ws_id = 1,  ws_type = 20;
    Pint       i, n;
    Pint       num_elems_appl_list, start_ind, err_ind,
               num_elems_impl_list;

    popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);
    popen_ws(ws_id, (void *)"/dev/tty", ws_type);

    /* Allocate space for n elements, where n is 12 */
    n = 12;
    num_elems_appl_list = n;
    line_inds.ints = (Pint *)calloc(n,sizeof(Pint));

    /* Start at index i, where i is 2 */
    i = 2;
    start_ind = i;

    ping_list_line_inds(ws_id, num_elems_appl_list, start_ind,
              &err_ind, &line_inds, &num_elems_impl_list);

    pclose_ws(ws_id);
    pclose_phigs();
}
```

After the call to `ping_list_line_inds()`, `line_inds` will contain a list of at most `n` polyline indices. The number of polyline indices returned by the implementation is given by `line_inds.num_ints`. The total number of polyline indices in the workstation is given by `num_elems_impl_list`.

D.2.1 Operation of `ping_list_line_inds`

The following steps take place during the execution of `ping_list_line_inds`.

1) When the inquiry function is called by the application, the parameters `num_elems_appl_list`, `line_inds`, `start_ind`, `*num_elems_impl_list` have the values as shown in Figure D.1.

2) The implementation copies the information from the implementation list to the application's list and stores the number of elements copied into the application's list in `line_inds->num_ints`.

   However, the number of list elements copied depends upon three variables: `n` the size of the application's list; `m` the size of the implementation's list and `i` the starting index of the copy. These variables yield three cases for determining the number of elements copied:

   a) `(m-i) <= 0`: This case means the starting index is greater than the number of elements in the implementation's list. No elements are copied into the application's list and the error indicator is set to error 2200.

   b) `(m-i) >= (n-1)`: In this case, the number of elements in the implementation's list (from the starting index to the end of the list) is greater than or equal to the number of elements the application has

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

allocated. So, `n` elements are copied and `line_inds->num_ints` is set to `n`; this case is illustrated in figure D.2.

c) `(m-i) < (n-1)`: In this case, the number of elements in the implementation's list (from the starting index to the end of the list) is less than the number of elements the application has allocated. So, `(m-i)` elements are copied and `line_inds->num_ints` is set to `(m-i)`; this case is illustrated in figure D.3.

3) The size of the implementation's list is stored in `num_elems_impl_list`

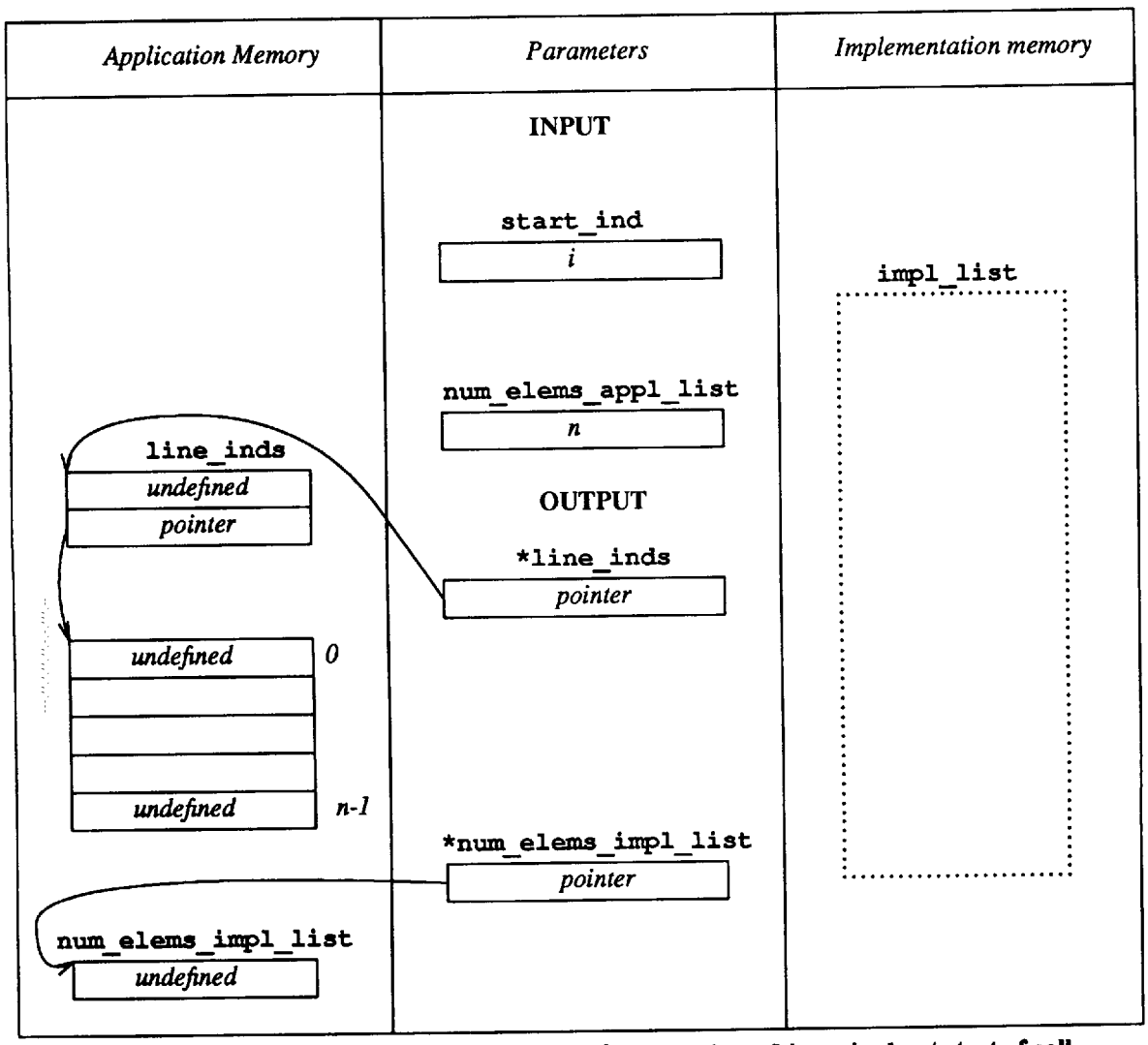


Figure D.1 — Values of parameters of `ping_list_line_inds` at start of call

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

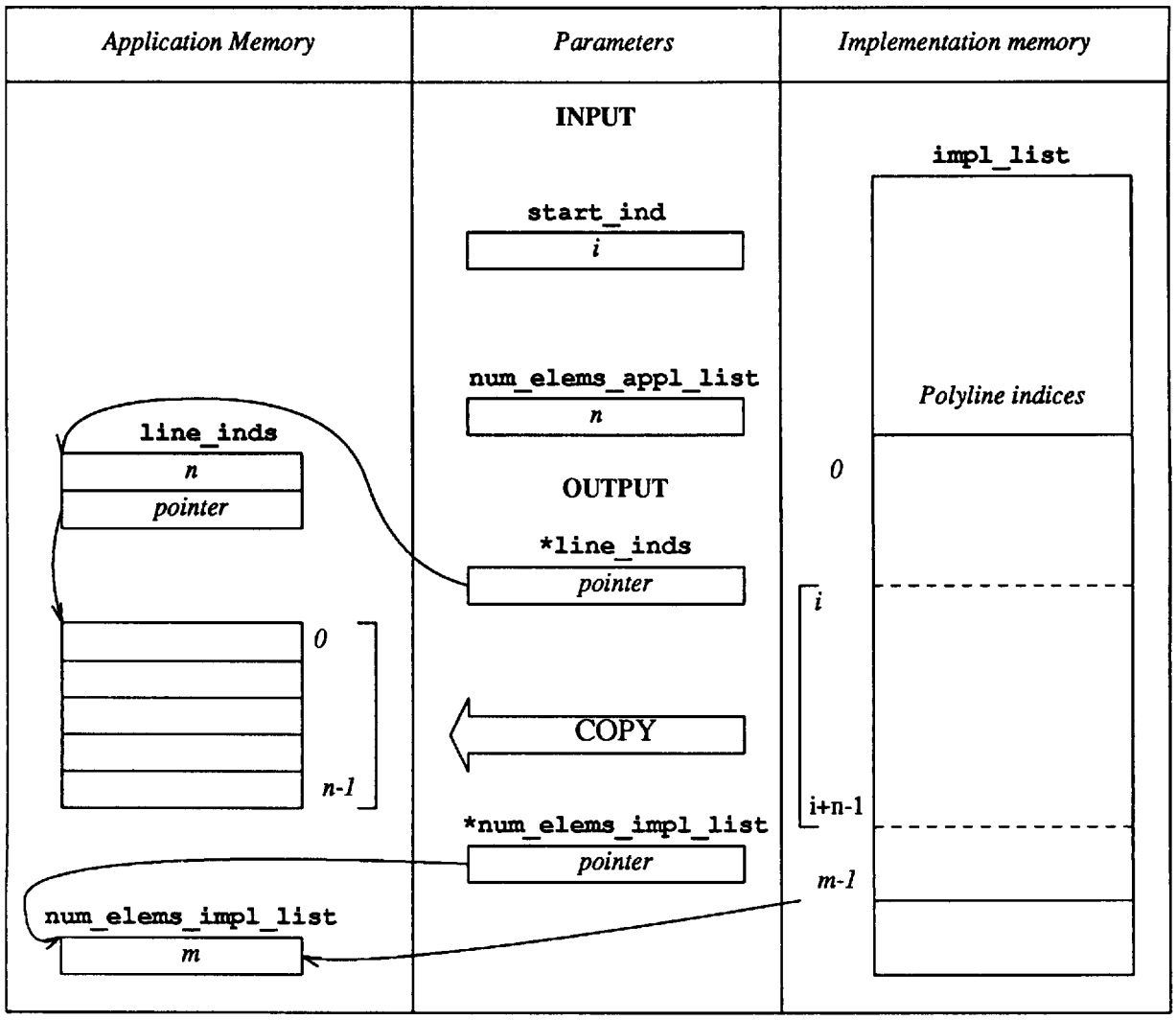


Figure D.2 — Values of parameters at end of call in case b

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

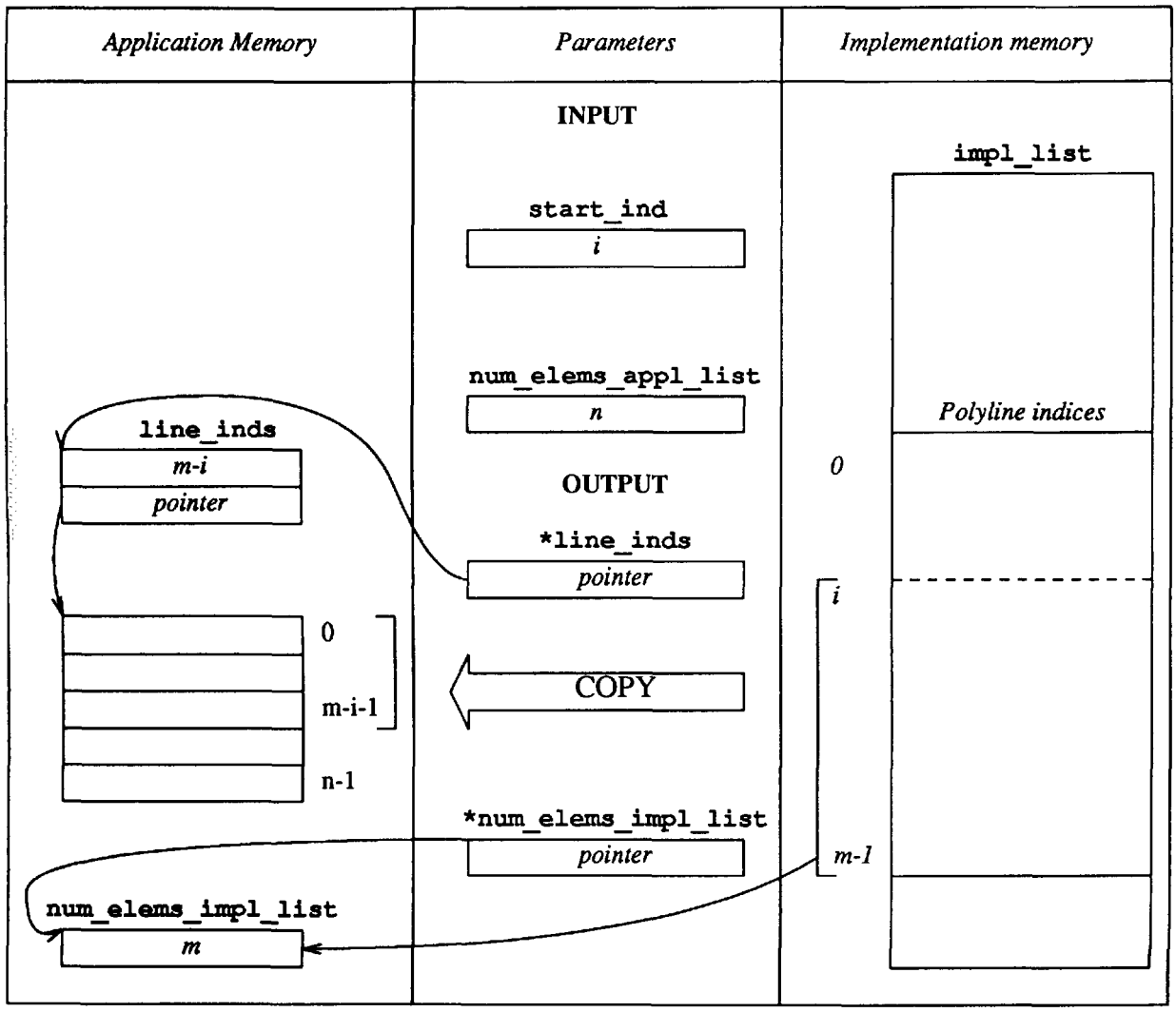


Figure D.3. Values of parameters at end of call in case c

D.3 Functions that return complex data structures

The auxiliary parameter `store` is used by functions which return complex data structures. The `store` parameter is a pointer to an implementation defined structure which the implementation uses to manage (that is, allocate and deallocate) the memory needed for returning information to the application.

This memory management policy is illustrated for the execution of the functions `pinq_stroke_st` and `pinq_pat_rep`. The sample program is in table D.2.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table D.2 — Sample program of the memory management policy for complex data structures

```
#include                "phigs.h"
main()
{
        Pint            ws_id = 1, ws_type = 20, stroke_dev = 1,
                        pat_ind = 2;
        Pint            err_ind;
        Pstore          store_1;
        Pstroke_data    *stroke_data;
        Ppoint_list     *init_stroke;
        Plimit          echo_area;
        Pop_mode        op_mode;
        Pecho_switch    echo_switch;
        Pint            init_view_ind, pet;

        Pstore          store_2;
        Ppat_rep        *pat_rep;

        popen_phigs(PDEF_ERR_FILE, PDEF_MEM_SIZE);;
        popen_ws(ws_id, (void *)"/dev/tty", ws_type);

        pcreate_store(&err_ind, &store_1);
        ping_stroke_st(ws_id, stroke_dev, PINQ_SET, store_1,
                       &err_ind, &op_mode, &echo_switch,
                       &init_view_ind, &init_stroke,
                       &pet, &echo_area, &stroke_data);
        pdel_store(&err_ind, &store_1);

        pcreate_store (&err_ind, &store_2);
        ping_pat_rep(ws_id, pat_ind, PINQ_SET, store_2, &err_ind,
                     &pat_rep);

        pdel_store(&err_ind, &store_2);

        pclose_ws(ws_id);
        pclose_phigs();
}
```

D.3.1 Operation of `pcreate_store`

The function `pcreate_store` allocates a store and returns a pointer to it to the application. At the start of a call, the parameters to this function have undefined values as illustrated in figure D.4.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

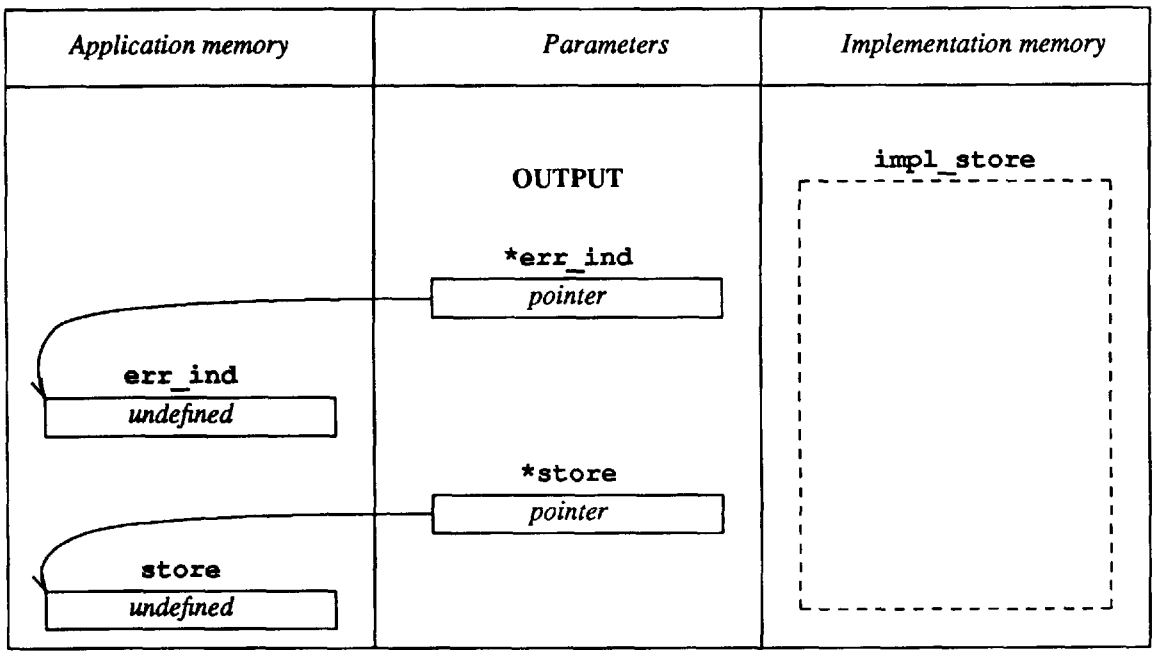


Figure D.4. Values of parameters of `pcreate_store` at start of call

As the result of a call, the implementation allocates the implementation store. The parameter `store` points to this memory; see figure D.5. After this, the implementation returns control to the application.

If the store cannot be allocated, the implementation returns abnormally to the application with the error indicator set to 2203.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

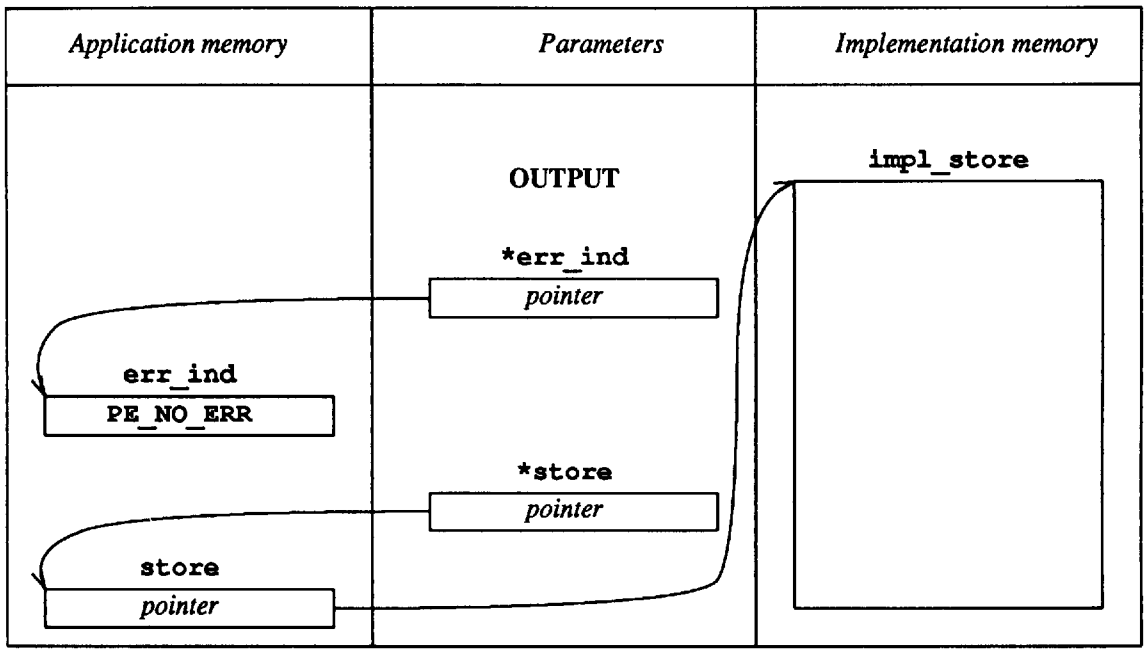


Figure D.5. Values of parameters of `pcreate_store` at end of successful call

D.3.2 Operation of `pinq_stroke_st` and `pinq_pat_rep`

In the example program, there are two stores created, `store_1` and `store_2`, and they are used in `pinq_stroke_st` and `pinq_pat_rep`, respectively. Two different stores are used to illustrate two uses of a store: `pinq_stroke_st` places two pieces of information (the stroke data and the initial stroke) into a single store; while `pinq_pat_rep` just places the pattern representation to a store. The same store could have been used in both functions but the sample program uses two to simplify the example.

The following steps take place.

1) The application calls `pinq_stroke_st`. The values of the relevant parameters are illustrated in figure D.6.

2) As a result of this call, the implementation places the information in the output parameters. The memory for the stroke data record and the initial stroke is managed by `store_1`. The addresses of the stroke data record and initial stroke are placed in the output parameters `*stroke_data` and `*init_stroke`, respectively. Figure D.7 illustrates this step.

   If the implementation cannot allocate enough memory to hold the stroke data record and the initial stroke, error number 2204 is returned via the error indicator and the call of `pinq_stroke_st` ends abnormally.

3) The application calls `pinq_pat_rep`. The values of the relevant parameters are illustrated in figure D.8.

4) As a result of call, the implementation stores the information in the output parameters. The memory for the pattern representation is managed by `store_2`. The address of the pattern representation is placed in the output parameter `*pat_rep`. Figure D.9.

   If the implementation cannot allocate enough memory to hold the the pattern representation, error number 2204 is returned via the error indicator and the call of `pinq_pat_rep` ends abnormally.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

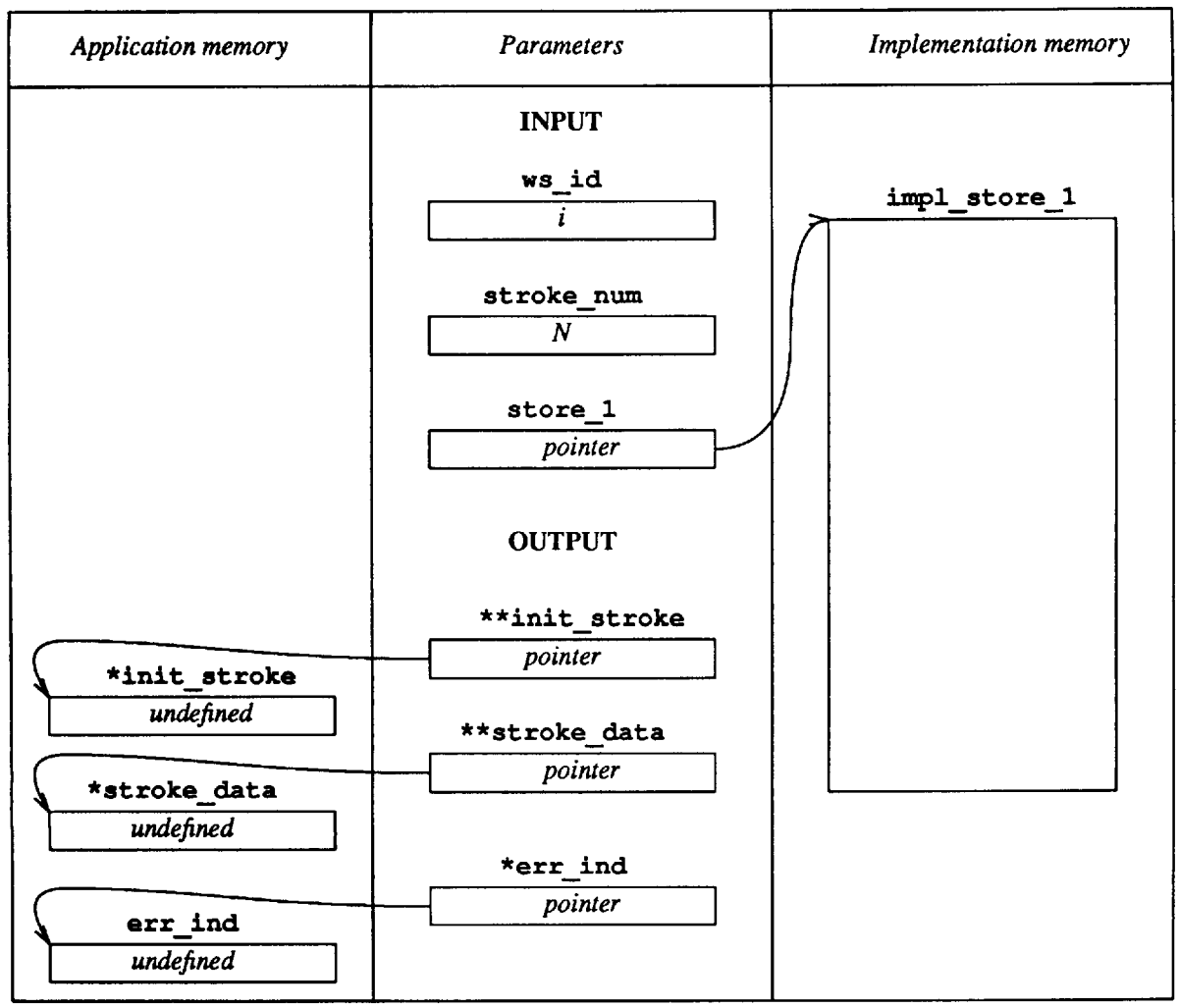


Figure D.6 — Values of parameters of `pinq_stroke_st` at start of call

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

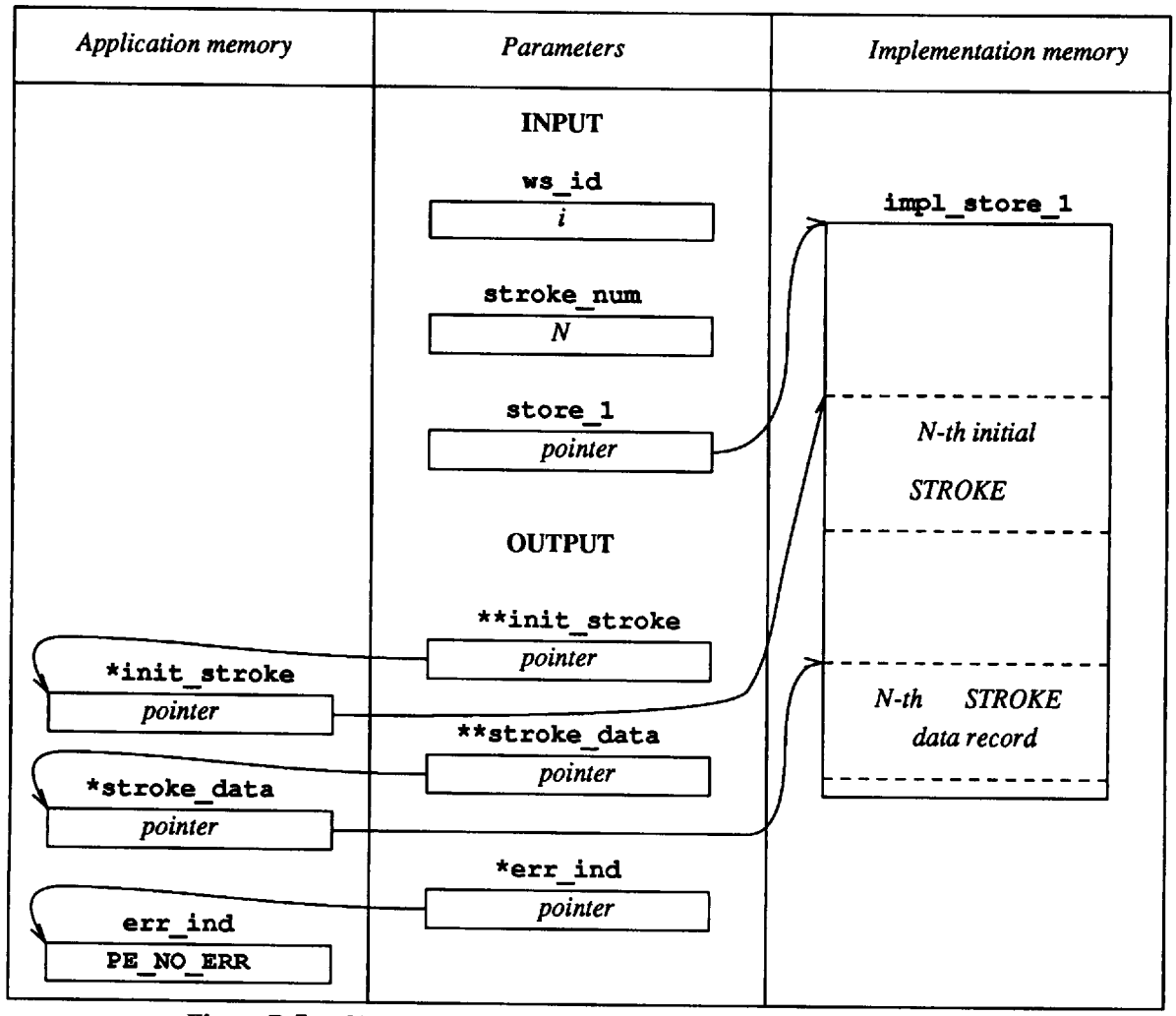


Figure D.7 — Values of parameters of `pinq_stroke_st` at end of call

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

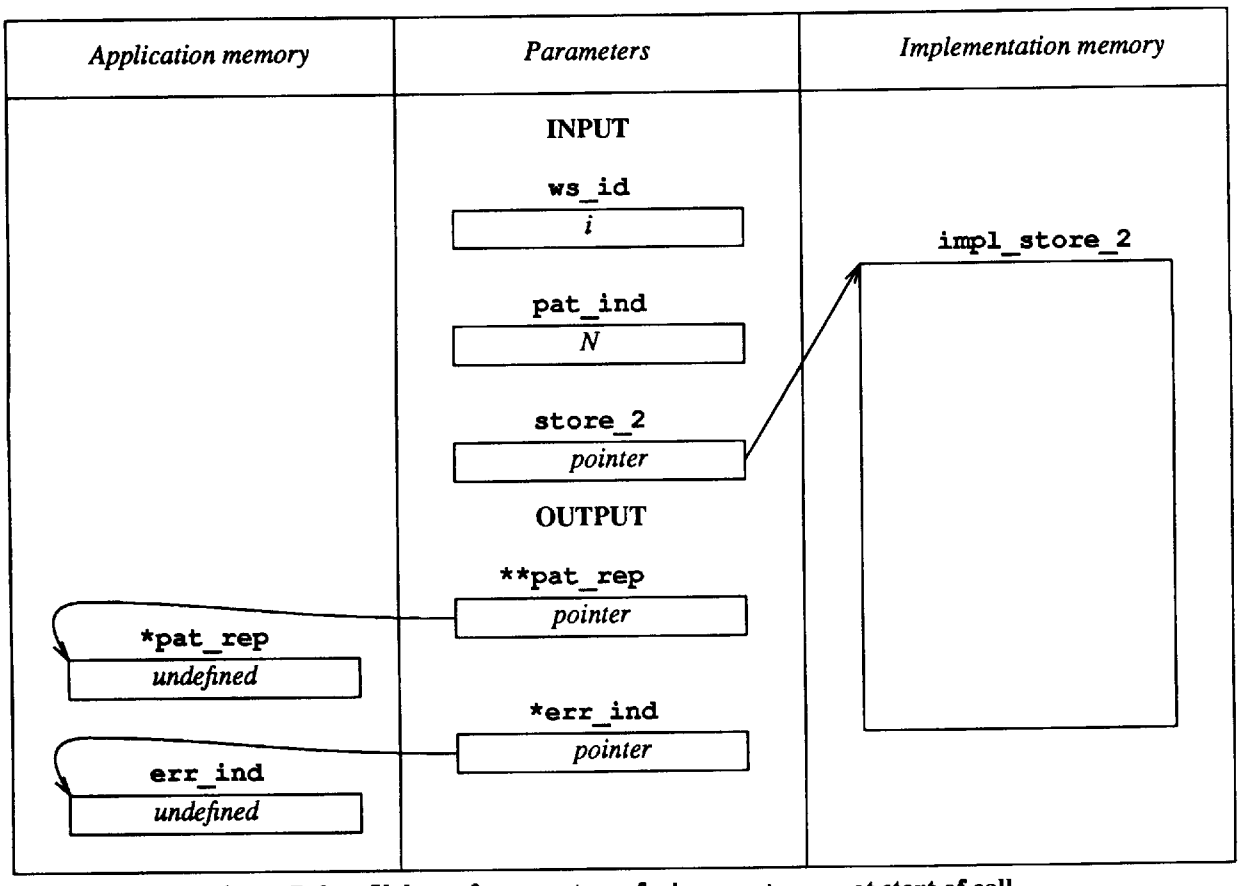


Figure D.8 — Values of parameters of `pinq_pat_rep` at start of call

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

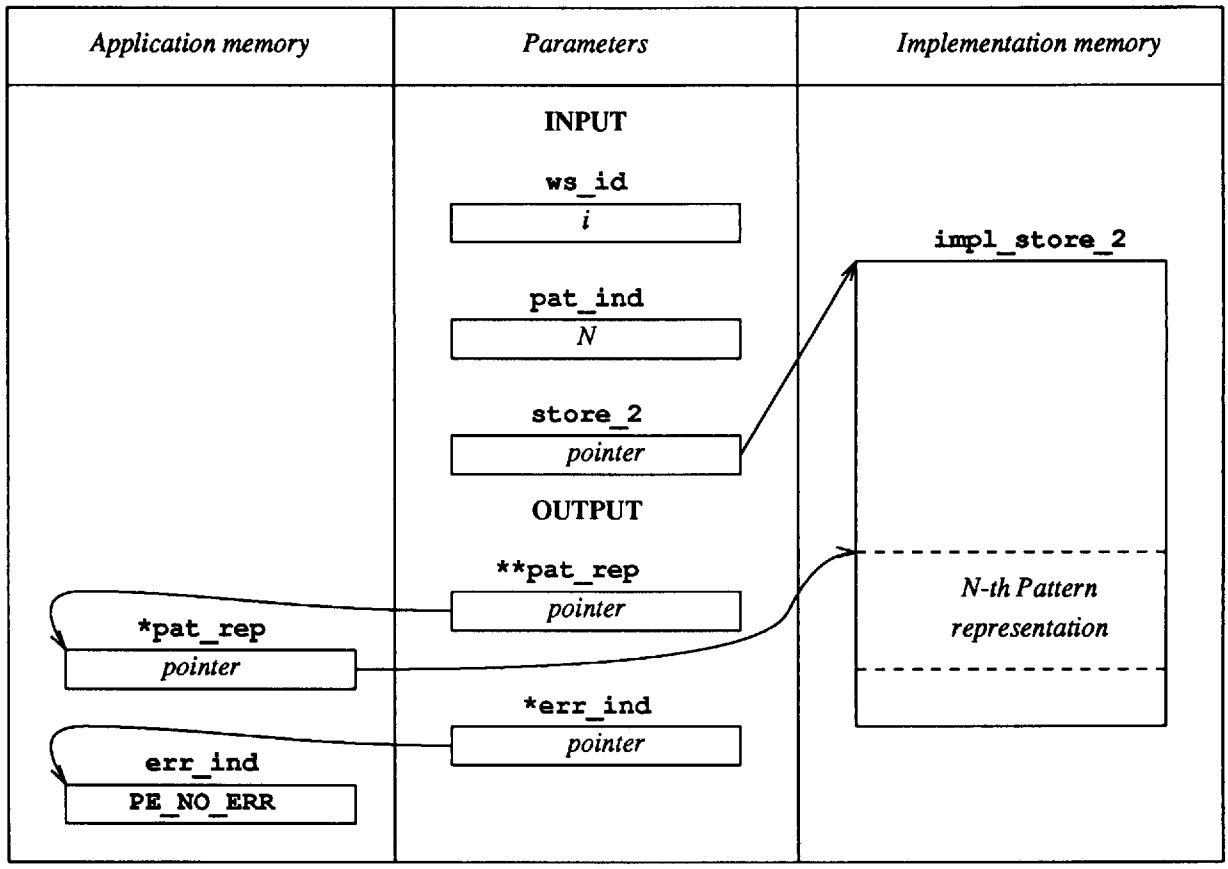


Figure D.9 — Values of parameters of `pinq_pat_rep` at end of call

D.3.3 Operation of `pdel_store`

The function frees `pdel_store` all memory that has been allocated by `pcreate_store` and all subsequent functions using a store.

The following actions take place.

1) The application calls `pdel_store`. The values of the parameters are illustrated in figure D.10.

2) As a result of this call, the implementation deallocates all memory referenced by the parameter store; furthermore, the parameter store is set to NULL; see figure D.11.

   If the deallocation fails, `pdel_store` ends abnormally and the error indicator is given an implementation dependent value.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

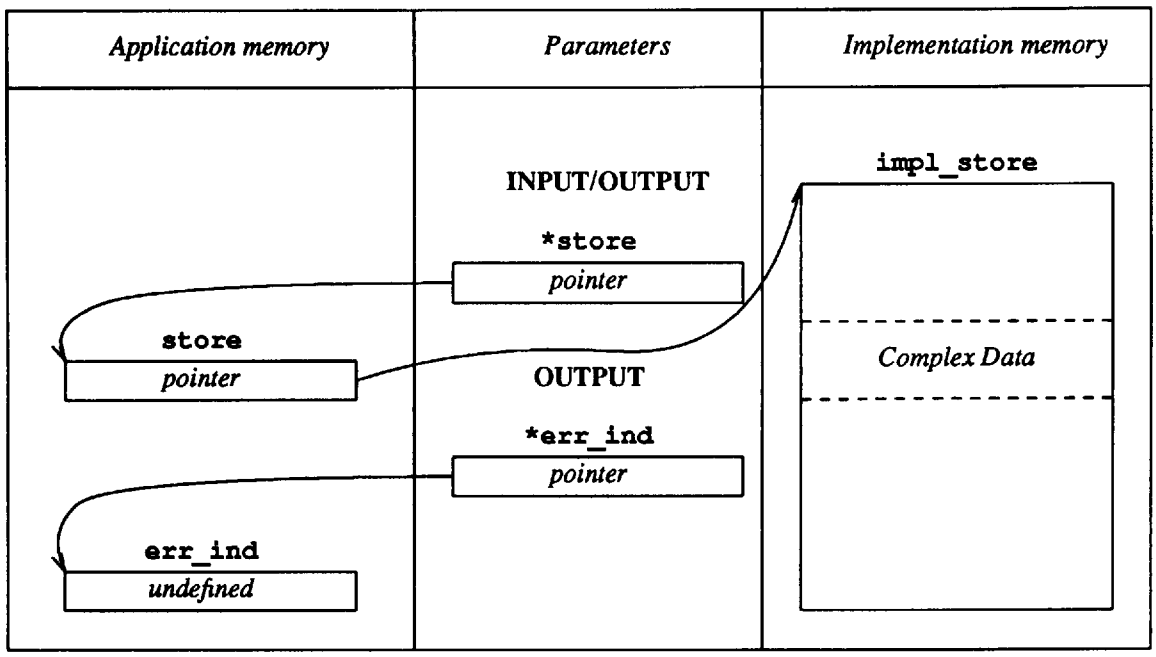


Figure D.10 — Values of parameters of `pdel_store` at start of call

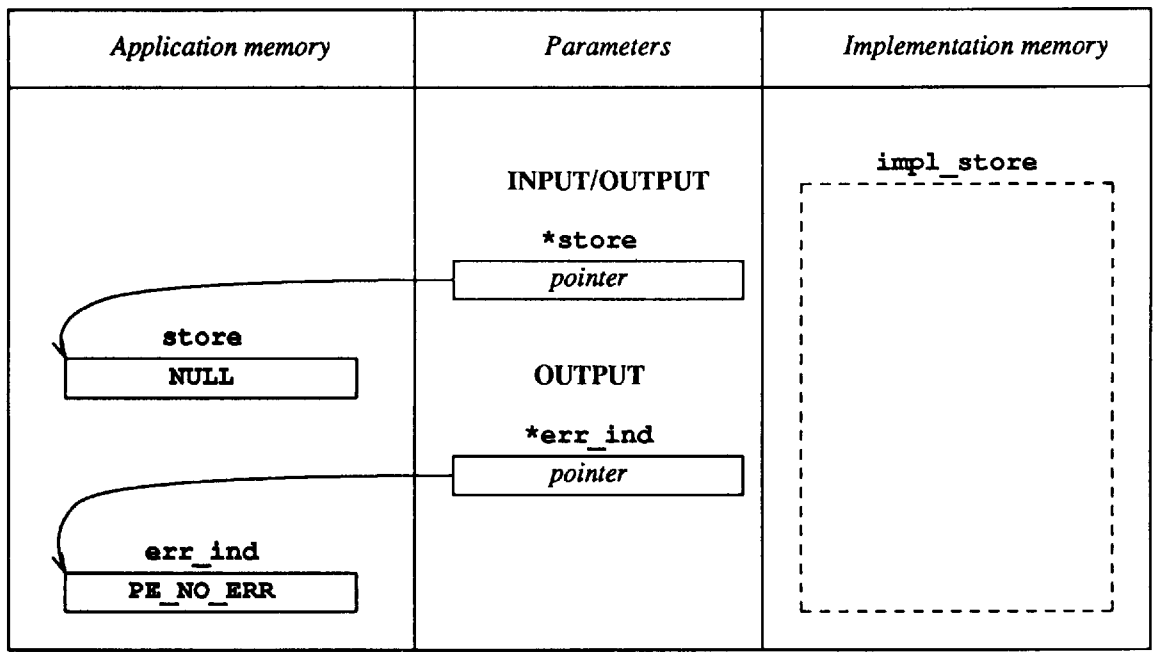


Figure D.11 — Values of parameters of `pdel_store` at end of a successful call

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577631 796 ■

Annex E

(informative)

Function Lists

Annex E lists the PHIGS functions.

E.1 List of functions ordered alphabetically by function name

ADD NAMES TO SET 94
ANNOTATION TEXT RELATIVE 88
ANNOTATION TEXT RELATIVE 3 88
APPLICATION DATA 105
ARCHIVE ALL STRUCTURES 109
ARCHIVE STRUCTURE NETWORKS 109
ARCHIVE STRUCTURES 109
AWAIT EVENT 120
BUILD TRANSFORMATION MATRIX 103
BUILD TRANSFORMATION MATRIX 3 103
CELL ARRAY 89
CELL ARRAY 3 89
CHANGE STRUCTURE IDENTIFIER 107
CHANGE STRUCTURE IDENTIFIER AND REFERENCES 108
CHANGE STRUCTURE REFERENCES 107
CLOSE ARCHIVE FILE 109
CLOSE PHIGS 85
CLOSE STRUCTURE 105
CLOSE WORKSTATION 86
COMPOSE MATRIX 102
COMPOSE MATRIX 3 102
COMPOSE TRANSFORMATION MATRIX 103
COMPOSE TRANSFORMATION MATRIX 3 103
COPY ALL ELEMENTS FROM STRUCTURE 106
CREATE STORE 154
DELETE ALL STRUCTURES 107
DELETE ALL STRUCTURES FROM ARCHIVE 111
DELETE ELEMENT 106
DELETE ELEMENT RANGE 106
DELETE ELEMENTS BETWEEN LABELS 106
DELETE STORE 154
DELETE STRUCTURE 107
DELETE STRUCTURE NETWORK 107
DELETE STRUCTURE NETWORKS FROM ARCHIVE 111
DELETE STRUCTURES FROM ARCHIVE 111
ELEMENT SEARCH 152
EMERGENCY CLOSE PHIGS 153
EMPTY STRUCTURE 107
ERROR HANDLING 153
ERROR LOGGING 153
ESCAPE 154
EVALUATE VIEW MAPPING MATRIX 104
EVALUATE VIEW MAPPING MATRIX 3 104
EVALUATE VIEW ORIENTATION MATRIX 104
EVALUATE VIEW ORIENTATION MATRIX 3 104

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577632 622 ■

EXECUTE STRUCTURE 105
FILL AREA 88
FILL AREA 3 88
FILL AREA SET 88
FILL AREA SET 3 88
FLUSH DEVICE EVENTS 120
GENERALIZED DRAWING PRIMITIVE 89
GENERALIZED DRAWING PRIMITIVE 3 89
GENERALIZED STRUCTURE ELEMENT 105
GET CHOICE 121
GET ITEM TYPE FROM METAFILE 122
GET LOCATOR 120
GET LOCATOR 3 120
GET PICK 121
GET STRING 121
GET STROKE 121
GET STROKE 3 120
GET VALUATOR 121
INCREMENTAL SPATIAL SEARCH 152
INCREMENTAL SPATIAL SEARCH 3 152
INITIALIZE CHOICE 113
INITIALIZE CHOICE 3 113
INITIALIZE LOCATOR 112
INITIALIZE LOCATOR 3 112
INITIALIZE PICK 114
INITIALIZE PICK 3 114
INITIALIZE STRING 114
INITIALIZE STRING 3 114
INITIALIZE STROKE 113
INITIALIZE STROKE 3 112
INITIALIZE VALUATOR 113
INITIALIZE VALUATOR 3 113
INQUIRE ALL CONFLICTING STRUCTURES 126
INQUIRE ANNOTATION FACILITIES 141
INQUIRE ARCHIVE FILES 125
INQUIRE ARCHIVE STATE VALUE 123
INQUIRE CHOICE DEVICE STATE 135
INQUIRE CHOICE DEVICE STATE 3 135
INQUIRE COLOUR FACILITIES 143
INQUIRE COLOUR MODEL 131
INQUIRE COLOUR MODEL FACILITIES 143
INQUIRE COLOUR REPRESENTATION 131
INQUIRE CONFLICT RESOLUTION 125
INQUIRE CONFLICTING STRUCTURES IN NETWORK 126
INQUIRE CURRENT ELEMENT CONTENT 150
INQUIRE CURRENT ELEMENT TYPE AND SIZE 150
INQUIRE DEFAULT CHOICE DEVICE DATA 148
INQUIRE DEFAULT CHOICE DEVICE DATA 3 147
INQUIRE DEFAULT DISPLAY UPDATE STATE 139
INQUIRE DEFAULT LOCATOR DEVICE DATA 146
INQUIRE DEFAULT LOCATOR DEVICE DATA 3 145
INQUIRE DEFAULT PICK DEVICE DATA 148
INQUIRE DEFAULT PICK DEVICE DATA 3 148
INQUIRE DEFAULT STRING DEVICE DATA 149

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577633 569 ■

INQUIRE DEFAULT STRING DEVICE DATA 3 149
INQUIRE DEFAULT STROKE DEVICE DATA 146
INQUIRE DEFAULT STROKE DEVICE DATA 3 146
INQUIRE DEFAULT VALUATOR DEVICE DATA 147
INQUIRE DEFAULT VALUATOR DEVICE DATA 3 147
INQUIRE DISPLAY SPACE SIZE 138
INQUIRE DISPLAY SPACE SIZE 3 138
INQUIRE DISPLAY UPDATE STATE 128
INQUIRE DYNAMICS OF STRUCTURES 145
INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES 139
INQUIRE EDGE FACILITIES 142
INQUIRE EDGE REPRESENTATION 130
INQUIRE EDIT MODE 125
INQUIRE ELEMENT CONTENT 151
INQUIRE ELEMENT POINTER 150
INQUIRE ELEMENT TYPE AND SIZE 150
INQUIRE ERROR HANDLING MODE 153
INQUIRE GENERALIZED DRAWING PRIMITIVE 144
INQUIRE GENERALIZED DRAWING PRIMITIVE 3 144
INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES 124
INQUIRE HIGHLIGHTING FILTER 132
INQUIRE HLHSR FACILITIES 138
INQUIRE HLHSR IDENTIFIER FACILITIES 138
INQUIRE HLHSR MODE 127
INQUIRE HLHSR MODE FACILITIES 138
INQUIRE INPUT QUEUE OVERFLOW 153
INQUIRE INTERIOR FACILITIES 141
INQUIRE INTERIOR REPRESENTATION 130
INQUIRE INVISIBILITY FILTER 132
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 144
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3 143
INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS 144
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES 123
INQUIRE LIST OF COLOUR INDICES 131
INQUIRE LIST OF EDGE INDICES 130
INQUIRE LIST OF INTERIOR INDICES 129
INQUIRE LIST OF PATTERN INDICES 130
INQUIRE LIST OF POLYLINE INDICES 128
INQUIRE LIST OF POLYMARKER INDICES 128
INQUIRE LIST OF TEXT INDICES 129
INQUIRE LIST OF VIEW INDICES 127
INQUIRE LOCATOR DEVICE STATE 133
INQUIRE LOCATOR DEVICE STATE 3 133
INQUIRE MODELLING CLIPPING FACILITIES 124
INQUIRE MORE SIMULTANEOUS EVENTS 126
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES 145
INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED 145
INQUIRE OPEN STRUCTURE 150
INQUIRE PATHS TO ANCESTORS 151
INQUIRE PATHS TO DESCENDANTS 151
INQUIRE PATTERN FACILITIES 142
INQUIRE PATTERN REPRESENTATION 131
INQUIRE PHIGS FACILITIES 124
INQUIRE PICK DEVICE STATE 136

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577634 4T5 ■

INQUIRE PICK DEVICE STATE 3 136
INQUIRE POLYLINE FACILITIES 139
INQUIRE POLYLINE REPRESENTATION 128
INQUIRE POLYMARKER FACILITIES 140
INQUIRE POLYMARKER REPRESENTATION 129
INQUIRE POSTED STRUCTURES 127
INQUIRE PREDEFINED COLOUR REPRESENTATION 143
INQUIRE PREDEFINED EDGE REPRESENTATION 142
INQUIRE PREDEFINED INTERIOR REPRESENTATION 142
INQUIRE PREDEFINED PATTERN REPRESENTATION 142
INQUIRE PREDEFINED POLYLINE REPRESENTATION 140
INQUIRE PREDEFINED POLYMARKER REPRESENTATION 140
INQUIRE PREDEFINED TEXT REPRESENTATION 141
INQUIRE PREDEFINED VIEW REPRESENTATION 139
INQUIRE SET OF OPEN WORKSTATIONS 125
INQUIRE SET OF WORKSTATIONS TO WHICH POSTED 149
INQUIRE STRING DEVICE STATE 137
INQUIRE STRING DEVICE STATE 3 137
INQUIRE STROKE DEVICE STATE 134
INQUIRE STROKE DEVICE STATE 3 134
INQUIRE STRUCTURE IDENTIFIERS 125
INQUIRE STRUCTURE STATE VALUE 123
INQUIRE STRUCTURE STATUS 151
INQUIRE SYSTEM STATE VALUE 123
INQUIRE TEXT EXTENT 141
INQUIRE TEXT FACILITIES 140
INQUIRE TEXT REPRESENTATION 129
INQUIRE VALUATOR DEVICE STATE 135
INQUIRE VALUATOR DEVICE STATE 3 135
INQUIRE VIEW FACILITIES 138
INQUIRE VIEW REPRESENTATION 127
INQUIRE WORKSTATION CATEGORY 137
INQUIRE WORKSTATION CLASSIFICATION 139
INQUIRE WORKSTATION CONNECTION AND TYPE 126
INQUIRE WORKSTATION STATE TABLE LENGTHS 145
INQUIRE WORKSTATION STATE VALUE 123
INQUIRE WORKSTATION TRANSFORMATION 132
INQUIRE WORKSTATION TRANSFORMATION 3 132
INTERPRET ITEM 122
LABEL 105
MESSAGE 86
OFFSET ELEMENT POINTER 106
OPEN ARCHIVE FILE 108
OPEN PHIGS 85
OPEN STRUCTURE 105
OPEN WORKSTATION 86
POLYLINE 87
POLYLINE 3 87
POLYMARKER 87
POLYMARKER 3 87
POST STRUCTURE 108
READ ITEM FROM METAFILE 122
REDRAW ALL STRUCTURES 86
REMOVE NAMES FROM SET 94

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

REQUEST CHOICE 117
REQUEST LOCATOR 116
REQUEST LOCATOR 3 116
REQUEST PICK 117
REQUEST STRING 118
REQUEST STROKE 117
REQUEST STROKE 3 116
REQUEST VALUATOR 117
RESTORE MODELLING CLIPPING VOLUME 98
RETRIEVE ALL STRUCTURES 110
RETRIEVE PATHS TO ANCESTORS 110
RETRIEVE PATHS TO DESCENDANTS 110
RETRIEVE STRUCTURE IDENTIFIERS 109
RETRIEVE STRUCTURE NETWORKS 110
RETRIEVE STRUCTURES 110
ROTATE 102
ROTATE X 101
ROTATE Y 101
ROTATE Z 101
SAMPLE CHOICE 119
SAMPLE LOCATOR 118
SAMPLE LOCATOR 3 118
SAMPLE PICK 119
SAMPLE STRING 119
SAMPLE STROKE 119
SAMPLE STROKE 3 118
SAMPLE VALUATOR 119
SCALE 101
SCALE 3 101
SET ANNOTATION STYLE 93
SET ANNOTATION TEXT ALIGNMENT 92
SET ANNOTATION TEXT CHARACTER HEIGHT 92
SET ANNOTATION TEXT CHARACTER UP VECTOR 92
SET ANNOTATION TEXT PATH 92
SET CHARACTER EXPANSION FACTOR 91
SET CHARACTER HEIGHT 91
SET CHARACTER SPACING 91
SET CHARACTER UP VECTOR 92
SET CHOICE MODE 115
SET COLOUR MODEL 97
SET COLOUR REPRESENTATION 96
SET CONFLICT RESOLUTION 109
SET DISPLAY UPDATE STATE 86
SET EDGE COLOUR INDEX 94
SET EDGE FLAG 93
SET EDGE INDEX 90
SET EDGE REPRESENTATION 96
SET EDGETYPE 93
SET EDGEWIDTH SCALE FACTOR 93
SET EDIT MODE 105
SET ELEMENT POINTER 106
SET ELEMENT POINTER AT LABEL 106
SET ERROR HANDLING 154
SET ERROR HANDLING MODE 154

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577636 278 ■

SET GLOBAL TRANSFORMATION 98
SET GLOBAL TRANSFORMATION 3 98
SET HIGHLIGHTING FILTER 96
SET HLHSR IDENTIFIER 97
SET HLHSR MODE 97
SET INDIVIDUAL ASF 95
SET INTERIOR COLOUR INDEX 93
SET INTERIOR INDEX 90
SET INTERIOR REPRESENTATION 95
SET INTERIOR STYLE 93
SET INTERIOR STYLE INDEX 93
SET INVISIBILITY FILTER 96
SET LINETYPE 90
SET LINEWIDTH SCALE FACTOR 90
SET LOCAL TRANSFORMATION 97
SET LOCAL TRANSFORMATION 3 97
SET LOCATOR MODE 115
SET MARKER SIZE SCALE FACTOR 90
SET MARKER TYPE 90
SET MODELLING CLIPPING INDICATOR 98
SET MODELLING CLIPPING VOLUME 98
SET MODELLING CLIPPING VOLUME 3 98
SET PATTERN REFERENCE POINT 94
SET PATTERN REFERENCE POINT AND VECTORS 94
SET PATTERN REPRESENTATION 96
SET PATTERN SIZE 94
SET PICK FILTER 111
SET PICK IDENTIFIER 111
SET PICK MODE 115
SET POLYLINE COLOUR INDEX 90
SET POLYLINE INDEX 89
SET POLYLINE REPRESENTATION 95
SET POLYMARKER COLOUR INDEX 91
SET POLYMARKER INDEX 89
SET POLYMARKER REPRESENTATION 95
SET STRING MODE 116
SET STROKE MODE 115
SET TEXT ALIGNMENT 92
SET TEXT COLOUR INDEX 91
SET TEXT FONT 91
SET TEXT INDEX 90
SET TEXT PATH 92
SET TEXT PRECISION 91
SET TEXT REPRESENTATION 95
SET VALUATOR MODE 115
SET VIEW INDEX 99
SET VIEW REPRESENTATION 99
SET VIEW REPRESENTATION 3 99
SET VIEW TRANSFORMATION INPUT PRIORITY 99
SET WORKSTATION VIEWPORT 100
SET WORKSTATION VIEWPORT 3 100
SET WORKSTATION WINDOW 100
SET WORKSTATION WINDOW 3 99
TEXT 87

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577637 104 ■

TEXT 3 87
TRANSFORM POINT 102
TRANSFORM POINT 3 102
TRANSLATE 100
TRANSLATE 3 100
UNPOST ALL STRUCTURES 108
UNPOST STRUCTURE 108
UPDATE WORKSTATION 86
WRITE ITEM TO METAFILE 122

E.2 List of functions ordered alphabetically by bound name

padd_names_set 94
panno_text_rel 88
panno_text_rel3 88
pappl_data 105
par_all_structs 109
par_struct_nets 109
par_structs 109
pawait_event 120
pbuild_tran_matrix 103
pbuild_tran_matrix3 103
pcell_array 89
pcell_array3 89
pchange_struct_id 107
pchange_struct_id_refs 108
pchange_struct_refs 107
pclose_ar_file 109
pclose_phigs 85
pclose_struct 105
pclose_ws 86
pcompose_matrix 102
pcompose_matrix3 102
pcompose_tran_matrix 103
pcompose_tran_matrix3 103
pcopy_all_elems_struct 106
pcreate_store 154
pdel_all_structs 107
pdel_all_structs_ar 111
pdel_elem 106
pdel_elem_range 106
pdel_elems_labels 106
pdel_store 154
pdel_struct 107
pdel_struct_net 107
pdel_struct_nets_ar 111
pdel_structs_ar 111
pelem_search 152
pemergency_close_phigs 153
pempty_struct 107
perr_hand 153
perr_log 153
pescape 154
peval_view_map_matrix 104

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577638 040 ■

peval_view_map_matrix3 104
peval_view_ori_matrix 104
peval_view_ori_matrix3 104
pexec_struct 105
pfill_area 88
pfill_area3 88
pfill_area_set 88
pfill_area_set3 88
pflush_events 120
pgdp 89
pgdp3 89
pget_choice 121
pget_item_type 122
pget_loc 120
pget_loc3 120
pget_pick 121
pget_string 121
pget_stroke 121
pget_stroke3 120
pget_val 121
pgse 105
pincr_spa_search 152
pincr_spa_search3 152
pinit_choice 113
pinit_choice3 113
pinit_loc 112
pinit_loc3 112
pinit_pick 114
pinit_pick3 114
pinit_string 114
pinit_string3 114
pinit_stroke 113
pinit_stroke3 112
pinit_val 113
pinit_val3 113
pinq_all_conf_structs 126
pinq_anno_facs 141
pinq_ar_files 125
pinq_ar_st 123
pinq_choice_st 135
pinq_choice_st3 135
pinq_colr_facs 143
pinq_colr_model 131
pinq_colr_model_facs 143
pinq_colr_rep 131
pinq_conf_res 125
pinq_conf_structs_net 126
pinq_cur_elem_content 150
pinq_cur_elem_type_size 150
pinq_def_choice_data 148
pinq_def_choice_data3 147
pinq_def_disp_upd_st 139
pinq_def_loc_data 146
pinq_def_loc_data3 145

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

pinq_def_pick_data 148
pinq_def_pick_data3 148
pinq_def_string_data 149
pinq_def_string_data3 149
pinq_def_stroke_data 146
pinq_def_stroke_data3 146
pinq_def_val_data 147
pinq_def_val_data3 147
pinq_disp_space_size 138
pinq_disp_space_size3 138
pinq_disp_upd_st 128
pinq_dyns_structs 145
pinq_dyns_ws_attrs 139
pinq_edge_facs 142
pinq_edge_rep 130
pinq_edit_mode 125
pinq_elem_content 151
pinq_elem_ptr 150
pinq_elem_type_size 150
pinq_err_hand_mode 153
pinq_gdp 144
pinq_gdp3 144
pinq_gse_facs 124
pinq_highl_filter 132
pinq_hlhsr_id_facs 138
pinq_hlhsr_mode 127
pinq_hlhsr_mode_facs 138
pinq_in_overf 153
pinq_int_facs 141
pinq_int_rep 130
pinq_invis_filter 132
pinq_line_facs 139
pinq_line_rep 128
pinq_list_avail_gdp 144
pinq_list_avail_gdp3 143
pinq_list_avail_gse 144
pinq_list_avail_ws_types 123
pinq_list_colr_inds 131
pinq_list_edge_inds 130
pinq_list_int_inds 129
pinq_list_line_inds 128
pinq_list_marker_inds 128
pinq_list_pat_inds 130
pinq_list_text_inds 129
pinq_list_view_inds 127
pinq_loc_st 133
pinq_loc_st3 133
pinq_marker_facs 140
pinq_marker_rep 129
pinq_model_clip_facs 124
pinq_more_simult_events 126
pinq_num_avail_in 145
pinq_num_disp_pris 145
pinq_open_struct 150

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

pinq_open_wss 125
pinq_pat_facs 142
pinq_pat_rep 131
pinq_paths_ances 151
pinq_paths_descs 151
pinq_phigs_facs 124
pinq_pick_st 136
pinq_pick_st3 136
pinq_posted_structs 127
pinq_pred_colr_rep 143
pinq_pred_edge_rep 142
pinq_pred_int_rep 142
pinq_pred_line_rep 140
pinq_pred_marker_rep 140
pinq_pred_pat_rep 142
pinq_pred_text_rep 141
pinq_pred_view_rep 139
pinq_string_st 137
pinq_string_st3 137
pinq_stroke_st 134
pinq_stroke_st3 134
pinq_struct_ids 125
pinq_struct_st 123
pinq_struct_status 151
pinq_sys_st 123
pinq_text_extent 141
pinq_text_facs 140
pinq_text_rep 129
pinq_val_st 135
pinq_val_st3 135
pinq_view_facs 138
pinq_view_rep 127
pinq_ws_cat 137
pinq_ws_class 139
pinq_ws_conn_type 126
pinq_ws_st 123
pinq_ws_st_table 145
pinq_ws_tran 132
pinq_ws_tran3 132
pinq_wss_posted 149
pinterpret_item 122
plabel 105
pmessage 86
poffset_elem_ptr 106
popen_ar_file 108
popen_phigs 85
popen_struct 105
popen_ws 86
ppolyline 87
ppolyline3 87
ppolymarker 87
ppolymarker3 87
ppost_struct 108
pread_item 122

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577641 635 ■

predraw_all_structs 86
premove_names_set 94
preq_choice 117
preq_loc 116
preq_loc3 116
preq_pick 117
preq_string 118
preq_stroke 117
preq_stroke3 116
preq_val 117
prestore_model_clip_vol 98
pret_all_structs 110
pret_paths_ances 110
pret_paths_descs 110
pret_struct_ids 109
pret_struct_nets 110
pret_structs 110
protate 102
protate_x 101
protate_y 101
protate_z 101
psample_choice 119
psample_loc 118
psample_loc3 118
psample_pick 119
psample_string 119
psample_stroke 119
psample_stroke3 118
psample_val 119
pscale 101
pscale3 101
pset_anno_align 92
pset_anno_char_ht 92
pset_anno_char_up_vec 92
pset_anno_path 92
pset_anno_style 93
pset_char_expan 91
pset_char_ht 91
pset_char_space 91
pset_char_up_vec 92
pset_choice_mode 115
pset_colr_model 97
pset_colr_rep 96
pset_conf_res 109
pset_disp_upd_st 86
pset_edge_colr_ind 94
pset_edge_flag 93
pset_edge_ind 90
pset_edge_rep 96
pset_edgetype 93
pset_edgewidth 93
pset_edit_mode 105
pset_elem_ptr 106
pset_elem_ptr_label 106

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

pset_err_hand 154
pset_err_hand_mode 154
pset_global_tran 98
pset_global_tran3 98
pset_highl_filter 96
pset_hlhsr_id 97
pset_hlhsr_mode 97
pset_indiv_asf 95
pset_int_colr_ind 93
pset_int_ind 90
pset_int_rep 95
pset_int_style 93
pset_int_style_ind 93
pset_invis_filter 96
pset_line_colr_ind 90
pset_line_ind 89
pset_line_rep 95
pset_linetype 90
pset_linewidth 90
pset_loc_mode 115
pset_local_tran 97
pset_local_tran3 97
pset_marker_colr_ind 91
pset_marker_ind 89
pset_marker_rep 95
pset_marker_size 90
pset_marker_type 90
pset_model_clip_ind 98
pset_model_clip_vol 98
pset_model_clip_vol3 98
pset_pat_ref_point 94
pset_pat_ref_point_vecs 94
pset_pat_rep 96
pset_pat_size 94
pset_pick_filter 111
pset_pick_id 111
pset_pick_mode 115
pset_string_mode 116
pset_stroke_mode 115
pset_text_align 92
pset_text_colr_ind 91
pset_text_font 91
pset_text_ind 90
pset_text_path 92
pset_text_prec 91
pset_text_rep 95
pset_val_mode 115
pset_view_ind 99
pset_view_rep 99
pset_view_rep3 99
pset_view_tran_in_pri 99
pset_ws_vp 100
pset_ws_vp3 100
pset_ws_win 100

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

ISO 9593 PT*4 91 ■ 4851903 0577643 408 ■

pset_ws_win3 99
ptext 87
ptext3 87
ptran_point 102
ptran_point3 102
ptranslate 100
ptranslate3 100
punpost_all_structs 108
punpost_struct 108
pupd_ws 86
pwrite_item 122

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

UDC 681.3.01:681.3.04(084):800.92

Descriptors: data processing, information interchange, graphic data processing, programming (computers), computer programs, programming languages, C (programming language).

Price based on 309 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale