

INTERNATIONAL STANDARD

ISO/IEC
9593-1

First edition
1990-06-01

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

Part 1: FORTRAN

*Systèmes de traitement de l'information — Infographie — Interfaces
langage entre un programme d'application et son support graphique —
Partie 1: FORTRAN*



Reference number
ISO/IEC 9593-1:1990(E)



Documents
Document Sharing Hub

Contents	Page
Introduction	v
1 Scope	1
2 Normative references	1
3 Principles	2
3.1 Specification	2
3.2 Mapping of PHIGS function names to FORTRAN subroutine names	2
3.3 Parameters	2
3.4 The FORTRAN subset	2
3.5 Error handling	3
4 Generating FORTRAN subroutine names	4
5 Data types	8
6 Enumeration types	17
7 List of the PHIGS function names	24
7.1 List of functions ordered alphabetically by bound name	24
7.2 List of functions ordered alphabetically by PHIGS function name	29
8 PHIGS errors specific to the FORTRAN binding	35
9 The PHIGS function interface	36
9.1 General principles	36
9.2 Control functions	37
9.3 Output primitive functions	39
9.4 Attribute specification functions	44
9.4.1 Bundled attribute selection	44
9.4.2 Individual attribute selection	45
9.4.3 Aspect source flag setting	50
9.4.4 Workstation attribute table definition	50
9.4.5 Workstation filter definition	52
9.4.6 Colour model control	53
9.4.7 HLHSR attributes	53
9.5 Transformation functions	54
9.5.1 Modelling transformations	54
9.5.2 View operations	55
9.5.3 Workstation transformation	56
9.5.4 Utility functions to support modelling	57
9.5.5 Utility functions to support viewing	62
9.6 Structure content functions	64
9.7 Structure manipulation functions	67

© ISO/IEC 1990
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

9.8	Structure display functions	69
9.9	Structure archiving functions	70
9.10	Input functions	74
9.10.1	Pick related structure elements	74
9.10.2	Initialization of input devices	74
9.10.3	Setting mode of input devices	87
9.10.4	Request input functions	89
9.10.5	Sample input functions	92
9.10.6	Event input functions	95
9.11	Metafile functions	98
9.12	Inquiry functions	100
9.12.1	Inquiry functions for operating state value	100
9.12.2	Inquiry functions for PHIGS description table	100
9.12.3	Inquiry functions for PHIGS state list	102
9.12.4	Inquiry functions for workstation state list	104
9.12.5	Inquiry functions for workstation description table	123
9.12.6	Inquiry functions for structure state list	144
9.12.7	Inquiry functions for structure content	144
9.12.8	Inquiry function for PHIGS error state list	169
9.13	Error control	171
9.14	Special interfaces	172
10	Utility functions not defined in PHIGS	173

Annexes

A	FORTRAN Examples	175
B	Function Lists	200
B.1	List of functions ordered alphabetically by function name	200
B.2	List of functions ordered alphabetically by bound name	206

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 9593-1 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 processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings* :

- *Part 1: FORTRAN*
- *Part 2: Extended Pascal*
- *Part 3: ADA*
- *Part 4: C*

Annex B forms an integral part of this part of ISO/IEC 9593. Annex A is for information only.

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 FORTRAN computer programming language.

This page intentionally left blank

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

Part 1: FORTRAN

1 Scope

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 FORTRAN language.

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 listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 1539 : 1980, *Information processing systems - Programming Languages - FORTRAN*.

ISO/IEC 9592-1 : 1989, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 1 - functional description*.

ISO/IEC TR 9973 : 1988, *Information processing - Procedures for registration of graphical items*.

3 Principles

3.1 Specification

This part of ISO/IEC 9593 defines the PHIGS language binding interface for FORTRAN 77, as described in ISO 1539 : 1980. With some minor modifications, application programs can be transported between full FORTRAN 77 and FORTRAN 77 Subset PHIGS installations.

3.2 Mapping of PHIGS function names to FORTRAN subroutine names

The function names of PHIGS are all mapped to FORTRAN subroutine names that start with the letter 'P'. The mapping is generally done in a one-to-one correspondence to functions defined in ISO/IEC 9592-1. However, some functions are split into more than one subroutine in this binding, due to the number of parameters required. The remaining letters after the first one are obtained by deriving a unique acronym from the words of the function name; e.g., OPEN becomes OP, WORKSTATION becomes WK. Hence, the FORTRAN subroutine name of PHIGS function OPEN WORKSTATION is POPWK. For a list of all abbreviations, see clause 4. Names used internally that may be known outside PHIGS, e.g., during linking, start with some easily recognized and documented form such as 'PH' (subroutine, function, and common block names). Therefore, no external names starting with this construct should be chosen when using PHIGS, in order to avoid name conflicts.

3.3 Parameters

In general, the order of PHIGS function parameters is preserved. For some subroutines, however, there are additional parameters that have been inserted in the normal parameter sequence (e.g., array length for arrays).

Values of input parameters are unaltered by any PHIGS function as well as PACK DATA RECORD and UNPACK DATA RECORD.

In order that any element of a list (member of a set), such as the set of structure names, can be inquired, in this binding the inquiry functions return only a single element of a list (member of a set). In addition, the total number of elements of the list (members of the set) is always returned. The elements (members) are numbered starting from 1; each invocation of the inquiry function requires the desired element (member) number as an input parameter and returns the corresponding element (member). When the list (set) is empty, a zero is returned as the number of elements (members) and the parameter representing the single element (member) in the list is undefined.

3.4 The FORTRAN subset

The binding for FORTRAN 77 Subset is different from that for full FORTRAN 77 in order to accommodate the FORTRAN 77 Subset restrictions.

Those PHIGS subroutines in the full FORTRAN 77 binding that have arguments of type CHARACTER(*) have alternative subroutine definitions that include fixed length character strings, CHARACTER*80, for the Subset.

In some cases an additional INTEGER parameter (the number of characters) appears in the parameter list and the Subset version is distinguished by the addition of a final 'S', so that they can coexist in the same implementation. In other cases the INTEGER is already present and the FORTRAN 77 Subset version has the same name as the full FORTRAN 77 version.

Principles

The FORTRAN subset

A full FORTRAN 77 implementation shall include both subroutines when the names are distinct and only the full FORTRAN 77 version when the names are the same.

The enumeration values in this binding may be redefined by replacing the PARAMETER statements with corresponding DATA statements.

3.5 Error handling

There are two error routines in every PHIGS system, named PERLOG and PERHND. The user may replace the latter with his/her own subroutine using the same name, PERHND, and calling sequence. Furthermore, this user-defined error routine may call the system-defined error logging procedure PERLOG.

4 Generating FORTRAN subroutine names

For the binding of the PHIGS functions that inquire lists (sets), the word ‘element’ (‘member’) is added to the PHIGS name.

The derivation of the abbreviation for the subroutine names is performed in several steps. First, plurals are reduced to their singular form, and then compound terms are reduced to maintain uniqueness and appropriate name length. Finally, each remaining word is replaced by the null string or by an abbreviation.

Table 1 - Reduction of plurals to singulars

DEVICES	→	DEVICE
ELEMENTS	→	ELEMENT
EVENTS	→	EVENT
FILES	→	FILE
IDENTIFIERS	→	IDENTIFIER
INDICES	→	INDEX
LABELS	→	LABEL
LENGTHS	→	LENGTH
NETWORKS	→	NETWORK
PATHS	→	PATH
PRIMITIVES	→	PRIMITIVE
PRIORITIES	→	PRIORITY
REFERENCES	→	REFERENCE
TRANSFORMATIONS	→	TRANSFORMATION
STRUCTURES	→	STRUCTURE
TYPES	→	TYPE
WORKSTATIONS	→	WORKSTATION

Generating FORTRAN subroutine names

Table 2 - Reduce compound terms for uniqueness

ANNOTATION TEXT CHARACTER HEIGHT	→	ATCH
ANNOTATION TEXT CHARACTER UP	→	ATCU
INQUIRE ANNOTATION TEXT	→	QAT
SET ANNOTATION TEXT	→	SAT
ANNOTATION TEXT RELATIVE	→	ATR
ARCHIVE ALL	→	ARA
ARCHIVE STRUCTURE IDENTIFIER	→	ASID
IDENTIFIER AND REFERENCE	→	IR
CHANGE STRUCTURE IDENTIFIER	→	CSTID
DELETE ALL STRUCTURE	→	DAS
DYNAMICS OF STRUCTURE	→	DSTR
DYNAMICS OF WORKSTATION ATTRIBUTES	→	DSWA
EDGE FLAG	→	EDFG
ELEMENT POINTER	→	EP
ELEMENT CONTENT	→	ECO
ELEMENT TYPE AND SIZE	→	ETS
ERROR HANDLING MODE	→	ERHM
EVALUATE VIEW MAPPING MATRIX	→	EVMM
EVALUATE VIEW ORIENTATION MATRIX	→	EVOM
GENERALIZED STRUCTURE ELEMENT	→	GSE
INDIVIDUAL ASF	→	IASF
LIST OF	→	E
E AVAILABLE GENERALIZED DRAWING PRIMITIVE 3	→	EGD3
MAXIMUM LENGTH	→	L
MODELLING CLIPPING VOLUME	→	MCV
MODELLING CLIPPING	→	MCL
PATTERN REFERENCE POINT AND VECTORS	→	PRPV
PATTERN REFERENCE POINT	→	PARF
RETRIEVE ALL	→	RA
RETRIEVE STRUCTURE IDENTIFIER	→	RSID
SET OF	→	element
STRUCTURE IDENTIFIER	→	SID
STRUCTURE PATH	→	STPA
STRUCTURE NETWORK	→	SN
STRUCTURE STATE	→	STRS
TEXT FONT	→	TXFN
TRANSFORM POINT	→	TP
UNPOST ALL	→	UPA
VIEW TRANSFORMATION	→	VT
WORKSTATION STATE VALUE	→	WKST
(FORTRAN 77 subset)	→	S

Generating FORTRAN subroutine names

Table 3 - Deletions

element	ALL	AND	AT	AVAILABLE
BETWEEN	DATA	DEVICE	EVENT	FACTOR
FROM	IN	LIST	METAFILE	MORE
NAMES	NUMBER	OF	SIZE	SUPPORTED
TABLE	TO	TYPE	VALUE	VECTOR
WHICH				

Table 4 - Abbreviations

3	→	3	EMPTY	→	EM
ADD	→	AD	ERROR	→	ER
ALIGNMENT	→	AL	ESCAPE	→	ESC
ANCESTORS	→	AN	EVALUATE	→	EV
ANNOTATION	→	AN	EXECUTE	→	EX
APPLICATION	→	AP	EXPANSION	→	XP
ARCHIVE	→	AR	EXTENT	→	X
AREA	→	A	FACILITIES	→	F
ARRAY	→	A	FILE	→	F
ASF	→	ASF	FILL	→	F
AWAIT	→	WAIT	FILTER	→	FT
BUILD	→	BL	FLAG	→	F
CATEGORY	→	CA	FLUSH	→	FLUSH
CELL	→	C	FONT	→	F
CHANGE	→	C	GENERALIZED	→	G
CHARACTER	→	CH	GET	→	GT
CHOICE	→	CH	GLOBAL	→	GM
CLASSIFICATION	→	CL	HANDLING	→	HND
CLOSE	→	CL	HEIGHT	→	H
COLOUR	→	C	HIGHLIGHTING	→	HL
COMPOSE	→	CO	HLHSR	→	HR
CONFLICT	→	CN	IDENTIFIER	→	ID
CONFLICTING	→	C	INCREMENTAL	→	I
CONNECTION	→	C	INDEX	→	I
CONTENT	→	CT	INDICATOR	→	I
COPY	→	C	INDIVIDUAL	→	I
CURRENT	→	C	INITIALIZE	→	IN
DEFAULT	→	D	INPUT	→	I
DESCENDANTS	→	DE	INQUIRE	→	Q
DELETE	→	D	INTERPRET	→	I
DISPLAY	→	D	INTERIOR	→	I
DRAWING	→	D	INVISIBILITY	→	IV
EDGE	→	ED	ITEM	→	ITM
EDGETYPE	→	EDT	LABEL	→	LB
EDGEWIDTH	→	EW	LENGTH	→	L
EDIT	→	ED	LINETYPE	→	LN
ELEMENT	→	EL	LINEWIDTH	→	LW
EMERGENCY	→	E	LOCAL	→	LM

Generating FORTRAN subroutine names

Table 4 (continued) - Abbreviations

LOCATOR	→	LC	RESOLUTION	→	RS
LOGGING	→	LOG	RESTORE	→	R
LOGICAL	→	L	RETRIEVE	→	RE
MAPPING	→	MP	ROTATE	→	RO
MARKER	→	MK	SAMPLE	→	SM
MATRIX	→	M	SCALE	→	SC
MESSAGE	→	MSG	SEARCH	→	S
MODE	→	M	SET	→	S
MODEL	→	MD	SIMULTANEOUS	→	SIM
NETWORK	→	N	SPACE	→	SP
OFFSET	→	OS	SPACING	→	SP
OPEN	→	OP	SPATIAL	→	S
ORIENTATION	→	OR	STATE	→	S
OVERFLOW	→	OV	STATUS	→	ST
PACK	→	P	STRING	→	ST
PATH	→	P	STROKE	→	SK
PATTERN	→	PA	STRUCTURE	→	ST
PHIGS	→	PH	STYLE	→	S
PICK	→	PK	SYSTEM	→	SY
POINTER	→	PT	TEXT	→	TX
POLYLINE	→	PL	TRANSFORM	→	T
POLYMARKER	→	PM	TRANSFORMATION	→	T
POST	→	PO	TRANSLATE	→	TR
POSTED	→	PO	UNPACK	→	U
PRECISION	→	PR	UNPOST	→	UPO
PREDEFINED	→	P	UP	→	UP
PRIMITIVE	→	P	UPDATE	→	U
PRIORITY	→	P	VALUATOR	→	VL
QUEUE	→	Q	VIEW	→	VW
RANGE	→	RA	VIEWPORT	→	V
READ	→	RD	VOLUME	→	VOL
RECORD	→	REC	WINDOW	→	W
REDRAW	→	R	WRITE	→	W
REFERENCE	→	RF	WORKSTATION	→	WK
REMOVE	→	RE	X	→	X
REPRESENTATION	→	R	Y	→	Y
REQUEST	→	RQ	Z	→	Z

5 Data types

In ISO/IEC 9592-1 parameters of several types are used. The following shows the correspondence between the types used in ISO/IEC 9592-1 and their realization in a FORTRAN implementation.

PHIGS Data Type	FORTRAN Data Types
I integer	INTEGER
A(I) array of integers	This is described more at the end of this clause, where the representations of CELL ARRAY and PATTERN are described.
R real	REAL
const $\times$ <i>simple_type</i> where <i>simple_type</i> is realized as I or R (vector of values, for example $2 \times R$) For input argument where const ≥ 6 or for output argument where const ≥ 4 , then array of constant length is used, otherwise use separate parameters.	
const 1 $\times$ const 2 $\times$ R (matrix of values, for example $2 \times 3 \times R$) REAL array (const 1, const 2) For example, in order to store the projection transformation defined by:	
	$x \rightarrow x'/w'$ $y \rightarrow y'/w'$ $z \rightarrow z'/w'$
	in a $4 \times 4 \times \text{REAL}$ matrix, the values shall be stored such that:
	$x' = p[1,1]*x + p[2,1]*y + p[3,1]*z + p[4,1]$ $y' = p[1,2]*x + p[2,2]*y + p[3,2]*z + p[4,2]$ $z' = p[1,3]*x + p[2,3]*y + p[3,3]*z + p[4,3]$ $w' = p[1,4]*x + p[2,4]*y + p[3,4]*z + p[4,4]$
S string	1) In a full FORTRAN 77 subroutine: a) INTEGER containing the number of characters returned (for output string argument only). b) CHARACTER*(*) containing the string. In addition, if a character string that is an input parameter may reasonably contain no characters, then an INTEGER (≥ 0) is used to give the number of characters to be passed to the subroutine. 2) In a FORTRAN 77 Subset subroutine: a) INTEGER containing the number of characters passed to the subroutine (for input string only, i.e. only one INTEGER needed for output). b) INTEGER containing the number of characters returned (for output string argument only). If the value is < 0 or > 80, error 2004 is

Data types

generated.

c) CHARACTER*80 containing the string.

P2 point REAL, REAL containing the X- and Y-values

const × P2 (only occurs in non-inquiry functions)

Separate REAL parameters, with the X- and Y- coordinates of one point being followed by the X- and Y- coordinates of the next.

P3 point REAL, REAL, REAL containing the X-, Y-, and Z-values

const × P3

If $\text{const} \geq 2$ REAL arrays $\text{xa}(\text{const})$, $\text{ya}(\text{const})$, $\text{za}(\text{const})$ are used, otherwise separate REAL arrays are used, with the X-, Y-, and Z-coordinates of one point being followed by the X-, Y-, and Z-coordinates of the next.

L(L(P2/3)) list of point lists (for fill area sets)

The following description applies to both 2D and 3D point lists except that the PZA array is not present for 2D point lists. The arguments that specify the list of point lists are as follows:

INTEGER NPL

INTEGER IXA(NPL)

REAL PXA(*), PYA(*), PZA(*)

where

NPL is the number of point lists

IXA is an array of end indices for the point lists

PXA, PYA, PZA are the coordinate arrays

The range of indices in PXA, PYA, and PZA of each point list is as follows:

1 to IXA(1) is the first point list

IXA(i-1)+1 to IXA(i) is the *i*th point list, for $i=2$, to NPL when $\text{NPL} \geq 2$

Thus, for example:

a) 1 is the start index for the 1st point list,

b) IXA(i-1)+1 is the start index for the *i*th point list for all $i=2$ to NPL when $\text{NPL} \geq 2$, and

c) IXA(i) is the end index for all point lists $i=1$ to NPL.

In the actual arguments specifying the list of point lists (supplied by the application program), the following conditions shall hold true; otherwise error 2005 is generated, with the allowable exception:

d) $\text{NPL} \geq 1$,

e) PXA, PYA and PZA are dimensioned by at least IXA(NPL) (however it is allowable for the implementation not to generate error 2005 in this case),

f) $\text{IXA}(1) \geq 3$ (the first point list is at least 3 points),

g) $\text{IXA}(i+1) - \text{IXA}(i) \geq 3$ for $i=1$ to $\text{NPL}-1$, when $\text{NPL} \geq 2$ (the *i*th point list is at least 3 points).

V2 2D vector

REAL, REAL containing the X- and Y-values specifying an offset from some reference point in the coordinate system of the reference point.

Data types

V3 3D vector	REAL, REAL, REAL containing the X-, Y-, and Z-values specifying an offset from some reference point in the coordinate system of the reference point.
E enumeration	INTEGER All values are mapped to the range zero to N-1, where N is the number of enumeration alternatives. Except for null values, the order of the enumeration alternatives is the same as in ISO/IEC 9592-1 : null values always appear in the first position. If the integer value given by the application program is not in the range 0 to N-1, there is a language binding error condition (error 2000).
const × E (only occurrence in PHIGS is const = 18)	An array of INTEGER elements of dimension const is used, each element being an enumeration alternative.
NM classification	INTEGER identifying the name of an element of a name set. The domain of the names is the set of INTEGERS from 0 to an implementation defined maximum. The minimum maximum is 63.
FR filter	A compound data type containing two name sets representing inclusion and exclusion sets. Represented as two sets. For input parameter: INTEGER giving the number of elements in the inclusion set, an array of INTEGERS giving the elements in the inclusion set, INTEGER giving the number of elements in the exclusion set, and an array of INTEGERS giving the elements in the exclusion set. For output parameter: INTEGER (input) giving the dimension (ni) of the array that will contain the elements of inclusion set. INTEGER (input) giving the dimension (nc) of the array that will contain the elements of exclusion set. INTEGER (output) giving the number of elements of the inclusion set array actually used. INTEGER (output) giving the number of elements of the exclusion set array actually used. Array of INTEGERS (output) containing the elements in the inclusion set. Array of INTEGERS (output) containing the elements in the exclusion set. If the size required to contain the inclusion or exclusion elements is greater than the size of the arrays supplied (as determined by the dimensions ni and ne), error 2001 will be generated and the value returned in the corresponding INTEGER (output) parameters will represent the input array size required.
L(FR) list of filter	This data type is only used in the incremental search functions and applies to both the "normal" and "inverted" lists of filters. The arguments that specify the list of filters are as follows: INTEGER FLN INTEGER FLISX(FLN) INTEGER FLIS(*) INTEGER FLESX(FLN) INTEGER FLES(*) where

Data types

FLN is the number filters

FLISX is an array of end indices of each inclusion set in the list filters

FLIS is the array of elements of the inclusion sets

FLESX is an array of end indices of each exclusion set in the list filters

FLES is the array of elements of the exclusion sets

The range of indices in FLIS of each inclusion set is as follows:

1 to FLISX(1) is the first inclusion set,

FLISX(i-1)+1 to FLISX(i) is the *i*th inclusion set,

for *i*=2, to FLN when $FLN \geq 2$

Correspondingly the range of indices in FLES of each exclusion set is as follows:

1 to FLESX(1) is the first exclusion set,

FLESX(i-1)+1 to FLESX(i) is the *i*th exclusion set,

for *i*=2, to FLN when $FLN \geq 2$

Thus, for example:

a) 1 is the start index for the first inclusion set,

b) FLISX(i-1)+1 is the start index for the *i*'th inclusion set for all *i*=2 to FLN when $FLN \geq 2$, and

c) FLISX(i) is the end index for all inclusion sets *i*=1 to FLN.

In the actual arguments specifying the list filters (supplied by the application program), the following conditions shall hold true; otherwise error 2006 is generated, with the allowable exception:

d) $FLN \geq 1$,

e) FLIS and FLES are dimensioned by at least FLESX(FLN) (however it is allowable for the implementation not to generate error 2006 in this case),

a set of *simple_type*

(for example, SET(NM) or SET(E)): INTEGER giving the number of elements in the set and an array of *simple_type* giving the elements in the set. The only direct use of the SET constructor involves either a classification, NM, or enumeration, E, as its *simple_type*; both are represented as INTEGERS.

PP pick path item

This data type is never independently accessed through the language binding interface, but is instead realized through the list of pick path items, or *pick path*, L(PP).

L(PP) pick path

INTEGER (input) giving the size of the INTEGER matrix within which the pick path will be returned, an INTEGER (output) returning the actual size of the pick path (only for output parameters) (however, if the size required is greater than the size of the input array size error 2001 will be generated and the value returned represents the input array size required), and a 3×N INTEGER matrix (output) returning the pick path, where the (1,*) components contain the structure identifiers, the (2,*) components contain the pick identifiers, and the (3,*) components contain the element sequence numbers.

ER element reference

This data type is never independently accessed through the language binding interface, but is instead realized through the list of element references, or *element reference path*, L(ER).

L(ER) element reference path

INTEGER (input) giving the size of the INTEGER matrix within which the element reference path will be returned, an INTEGER (output) returning the actual size of the element reference path (only for output parameters) (however, if the size required is greater than the size of the input array size error 2001 will be generated and the value returned represents the input array size required), and a $2 \times N$ INTEGER matrix (output) returning the element reference path, where the (1,*) components contain the structure identifiers, and the (2,*) components contain the the element sequence numbers.

L(L(ER)) list of element reference paths

For inquiry functions, a single call only returns a single element of the list. Each element of the list is a list itself as described under ELEMENT REFERENCE PATH above.

For a complete list of length n ,

- a) INTEGER (input parameter) containing the *sequence number* (see below) of the required list elements (in the range $0 \dots n$).
- b) INTEGER (output parameter) containing the number of items in the list, n .
- c) parameters of the type described under ELEMENT REFERENCE PATH.

If the sequence number given is 0, the requested element returned is undefined, but an error is not indicated thereby; the number of items in the list n is returned. If the sequence number given is <0 or $>n$, then error 2002 is indicated, the number of items in the list is returned, but the requested element is undefined; the exception to this occurs when the list size is 0, and in that case an error is not indicated thereby.

HS2 2D half-space

Two (2) REALs giving the X- and Y-components of a point in modelling coordinates and two (2) REALs giving the X- and Y-components of a normal vector in modelling coordinates. This data type is never independently accessed through the language binding interface, but is instead realized through the list of 2D half-spaces, L(HS2).

L(HS2) 2D half-space

2D half-spaces are only used as a component of a list, i.e., L(HS2), and are hence always specified as a component of a two dimensional array.

INTEGER NHALFS

REAL HALFSP(4,NHALFS)

where

NHALFS is the number of modelling clipping half spaces in the list, for the i th modelling clipping half-space:

HALFSP(1, i) is the X-component of the point,

HALFSP(2, i) is the Y-component of the point,

HALFSP(3, i) is the DX-component of the normal vector,

HALFSP(4, i) is the DY-component of the normal vector.

HS3 3D half-space

Three (3) REALs giving a point in modelling coordinates and three (3) REALs giving a normal vector in modelling coordinates. This data type is never independently accessed through the language binding interface, but is

Data types

instead realized through the list of 3D half-spaces, L(HS3).

L(HS3) 3D half-spaces 3D half-spaces are only used as a component of a list, i.e., L(HS3), and are hence always specified as a component of a two dimensional array.

INTEGER NHALFS

REAL HALFSP(6,NHALFS)

where

NHALFS is the number of modelling clipping half spaces in the list,
for the *i*th modelling clipping half-space:

HALFSP(1,*i*) is the X-component of the point,

HALFSP(2,*i*) is the Y-component of the point,

HALFSP(3,*i*) is the Z-component of the point,

HALFSP(4,*i*) is the DX-component of the normal vector,

HALFSP(5,*i*) is the DY-component of the normal vector,

HALFSP(6,*i*) is the DZ-component of the normal vector.

FP font/precision pair This data type is a specific case of the more general ordered pair of different types (described next).

an ordered pair of different types

(for example (I;E) or font/precision pair, FP)

The different types are represented in turn in the FORTRAN parameter list.

SE structure element This compound data type is realized directly as the constituent parts identified in ISO/IEC 9592-1. The data type of each of the constituent parts is described in this clause.

PS posted structure This compound data type is realized directly as the constituent parts identified in ISO/IEC 9592-1. The data type of each of the constituent parts is described in this clause.

B bounding range REAL, REAL specifying a range with the first value less than or equal to the second value.

const × B (for example 2×B or 3×B) For input argument where const ≥ 3 or for output argument where const ≥ 2, use array of constant length, otherwise, use separate parameters. For those cases where const = 2 for input arguments and the parameter list would have more than 9 arguments when applying this rule, use an array of constant length.

CLR colour specification For input parameter:
INTEGER giving the length (*n*) of the list of colour components dictated by the colour model. REAL array of at least dimension *n*. If the length of the list, as described by *n*, is inappropriate for the colour model in effect, error 118 is indicated.

For output parameter:

INTEGER (input) giving the dimension (*n*) of the array that will contain the list of colour components. INTEGER (output) giving the number of elements of the array actually used. REAL array of at least dimension *n*.

Data types

If the size required to contain the list of colour components is greater than the size of the arrays supplied (as determined by the dimension n), error 2001 will be generated and the value returned in the INTEGER (output) parameter will represent the input array size required.

CC chromaticity coefficient	This only occurs as $3 \times CC$ and is realized as a REAL array of nine (9) elements representing the primary colour chromaticity coefficients, (u' , v') and luminance value (Y).
C connection identifier	INTEGER where the set of valid values is implementation dependent.
F file	INTEGER where the set of valid values is implementation dependent.
W workstation type	INTEGER where the set of valid values is implementation dependent.
MCV modelling clipping volume	This data type is never used through the language binding interface, but is instead used to describe internal system state in ISO/IEC 9592-1.
G2 2D generalized drawing primitive identifier	INTEGER where the set of legal values is described in ISO/IEC 9592-1.
G3 3D generalized drawing primitive identifier	INTEGER where the set of legal values is described in ISO/IEC 9592-1.
GS generalized structure element identifier	INTEGER where the set of legal values is described in ISO/IEC 9592-1.
AI archive file identifier	INTEGER where an implementation may restrict the range but shall at least provide all non-negative integers that are available at that implementation.
PI pick identifier	INTEGER where an implementation may restrict the range but shall at least provide all non-negative integers that are available at that implementation. The default value for pick identifier is zero.
EI escape identifier	INTEGER where the set of legal values is described in ISO/IEC 9592-1.
FN function name	INTEGER where the range is shown in clause 6.
WI workstation identifier	INTEGER where an implementation may restrict the range but shall at least provide all non-negative integers that are available at that implementation.
D data record	Represented as a set of scalar values and an array of type CHARACTER*80 containing the data. In addition, an INTEGER input parameter is used to dimension the array. Where the data record is an output parameter, an additional output argument 'number of array elements of data record occupied' is needed. There are no scalar values except where the data record contains

Data types

values that are compulsory in PHIGS.

Special utility functions are defined to pack INTEGER, REAL, and CHARACTER data into the data record and to unpack the data record to the individual data items (PPREC, PUREC). The content of the packed data records is implementation dependent, but PPREC should perform the inverse function of PUREC and vice versa.

For INQUIRE CURRENT ELEMENT CONTENT and INQUIRE ELEMENT CONTENT the data record is returned directly into integer, real and character arrays.

list of n values of one *underlying_type* (for example L(I))

- 1) For input parameter:
 - a) INTEGER (input) containing length n of the list (unless the length is already present as a separate PHIGS parameter, in that case it is not duplicated)
 - b) array of dimension n , whose elements are of the appropriate *underlying_type*.

When the length can actually be defined as zero within PHIGS, the binding indicates the array dimension by *. The implementation checks that the given length is ≥ 1 .

- 2) For output parameter in non-inquiry functions:
 - a) INTEGER (input) containing the dimension of the array
 - b) INTEGER (output) containing the number of elements of the array actually used
 - c) an array whose elements are of the appropriate *underlying_type*. The input dimension being too small is a language binding error condition (error 2001).

In both cases (input or output), where the *underlying_type* is a 2D point, there is a REAL array for the X-coordinates and another for the Y-coordinates, and in the case of a 3D point, an additional REAL array for the Z-coordinates.

- 3) For inquiry functions, a single call only returns a single element of the list. For a complete list of length n ,
 - a) INTEGER (input) containing the *sequence number* (see below) of required list element (in the range $0 \dots n$).
 - b) INTEGER (output) containing the number of items in the list, n .
 - c) a parameter of the appropriate *underlying_type* containing the requested element.

If the sequence number given is 0, the requested element returned is undefined, but an error is not indicated thereby; the number of items in the list n is returned. If the sequence number given is < 0 or $> n$, then error 2002 is indicated, the number of items in the list is returned, but the requested element is undefined; the exception to this occurs when the list size is 0, and in that case an error is not indicated thereby.

- 4) A complete inquired list is returned from a single call when the maximum

Data types

size of the list is a small constant *m*:

- a) INTEGER (output) containing the number of elements of the array actually used.
- b) an array of dimension *m*, whose elements are of the appropriate *underlying_type*.

The representation of CELL ARRAY and PATTERN allows the user of the routines requiring a cell array parameter to pass any portion of the array as an argument. Two examples should make this clear.

Certainly the user can pass an entire two-dimensional array. In this case the number of columns of the cell array is the same as the dimensions of the FORTRAN array:

```
INTEGER DIMX, DIMY, CELLS (DIMX,DIMY)
CALL PCA (X1,Y1,X2,Y2,DIMX,DIMY,1,1,DIMX,DIMY,CELLS)
```

(1,1)	(2,1)	(3,1)	...	(DIMX,1)
(1,2)	(2,2)	(3,2)	...	(DIMX,2)
:	:	:		:
(1,DIMY)	(2,DIMY)	(3,DIMY)	...	(DIMX,DIMY)

To use an arbitrary portion of an array the user passes the upper left corner of the portion as starting address and the dimensions of the portion of interest of the array. The area inside the small box is the cell array being passed:

```
INTEGER STARTX, STARTY, DX, DY, DIMX, DIMY, CELLS (DIMX,DIMY)
DATA STARTX/3/, STARTY/6/, DX/2/, DY/3/
CALL PCA (X1,Y1,X2,Y2,DIMX,DIMY,STARTX,STARTY,DX,DY,CELLS)
```

(1,1)	(2,1)	(3,1)	(4,1)	...	(DIMX,1)
(1,2)	(2,2)	(3,2)	(4,2)	...	(DIMX,2)
:	:	:	:		:
:	:	:	:		:
(1,6)	(2,6)	(3,6)	(4,6)	...	(DIMX,6)
(1,7)	(2,7)	(3,7)	(4,7)	...	(DIMX,7)
(1,8)	(2,8)	(3,8)	(4,8)	...	(DIMX,8)
:	:	:	:		:
:	:	:	:		:
(1,DIMY)	(2,DIMY)	(3,DIMY)	(4,DIMY)	...	(DIMX,DIMY)

6 Enumeration types

All the enumeration types of PHIGS are mapped to FORTRAN INTEGERS. The correspondence between PHIGS scalars and FORTRAN INTEGERS is shown as follows in a list of symbolic FORTRAN constants that may be included in any application program. This clause contains a mapping PHIGS enumeration types to FORTRAN variable names. In a FORTRAN 77 Subset implementation, this mapping could be accomplished by the DATA statement.

"Line type", "marker type" and "colour model" are defined as INTEGER rather than enumeration types in PHIGS. Constant definitions for the explicitly defined and required values of these conceptually unbounded ranges are provided as a convenience.

Also, a numbering of all PHIGS functions is given for use in the error handling procedures.

Mnemonic FORTRAN names and their values for PHIGS ENUMERATION type values:

annotation style	unconnected, lead line using current polyline attributes
------------------	--

```

INTEGER      PUNCON,  PLDLN
PARAMETER( PUNCON=1, PLDLN=2 )

```

INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.

```

archive state      ARCL,          AROP
                   INTEGER  PARCL,  PAROP
                   PARAMETER( PARCL=0, PAROP=1 )

```

aspect	identifier					
	INTEGER	PLN,	PLWSC,	PPLCI,	PMK,	PMKSC,
1		PPMCI,	PTXFN,	PTXPR,	PCHXP,	PCHSP,
2		PTXCI,	PIS,	PISI,	PICI,	PEDFG,
3		PEDT,	PEWSC,	PEDCI		
	PARAMETER(	PLN=0,	PLWSC=1,	PPLCI=2,	PMK=3,	PMKSC=4,
1		PPMCI=5,	PTXFN=6,	PTXPR=7,	PCHXP=8,	PCHSP=9,
2		PTXCI=10,	PIS=11,	PISI=12,	PICI=13,	PEDFG=14,
3		PEDT=15,	PEWSC=16,	PEDCI=17)		

aspect	source	bundled,	individual
	INTEGER	PBUNDL,	PINDIV
	PARAMETER(	PBUNDL=0,	PINDIV=1)

clipping indicator	noclip,	clip
INTEGER	PNCLIP,	PCLIP
PARAMETER(	PNCLIP=0,	PCLIP=1)

```

colour available      monochrome, colour
                      INTEGER      PMONOC,  PCOLOR
                      PARAMETER( PMONOC=0, PCOLOR=1 )

```

```
colour model      RGB,       CIE,       HSV,       HLS
                  INTEGER    PRGB,      PCIE,      PHSV,      PHLS
                  PARAMETER( PRGB=1,   PCIE=2,   PHSV=3,   PHLS=4 )
```

INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.

Enumeration types

composition type	preconcatenate, INTEGER PCPRE, PARAMETER(PCPRE=0,	postconcatenate, PCPOST, PCPOST=1,	replace PCREPL PCREPL=2)
conflict resolution	maintain, INTEGER PCRMNT, PARAMETER(PCRMNT=0,	abandon, PCRABA, PCRABA=1,	update PCRUPD PCRUPD=2)
control flag	conditionally, INTEGER PCONDI, PARAMETER(PCONDI=0,	always PALWAY PALWAY=1)	
deferral mode	ASAP, INTEGER PASAP, PARAMETER(PASAP=0,	BNIG, PBNIG, PBNIG=1,	BNIL, PBNIL, PBNIL=2,
		ASTI, PASTI, PASTI=3,	WAIT PWAITD PWAITD=4)
device coordinate units	metres, INTEGER PMETRE, PARAMETER(PMETRE=0,	other POTHU POTHU=1)	
display surface empty	notempty, INTEGER PNEMPT, PARAMETER(PNEMPT=0,	empty PEMPTY PEMPTY=1)	
dynamic modification	IRG, INTEGER PIRG, PARAMETER(PIRG=0,	IMM, PIMM, PIMM=1,	CBS PCBS PCBS=2)
echo switch	noecho, INTEGER PNECHO, PARAMETER(PNECHO=0,	echo PECHO PECHO=1)	
edit mode	insert, INTEGER PINSRT, PARAMETER(PINSRT=0,	replace PREPLC PREPLC=1)	

Enumeration types

element type	all,	nil,	polyline 3,	polyline,...		
INTEGER(	PEALL,	PENIL,	PEPL3,	PEPL,		
1	PEPM3,	PEPM,	PETX3,	PETX,		
2	PEATR3,	PEATR,	PEFA3,	PEFA,		
3	PEFAS3,	PEFAS,	PECA3,	PECA,		
4	PEGDP3,	PEGDP,	PEPLI,	PEPMI,		
5	PETXI,	PEII,	PEEDI,	PELN,		
6	PELWSC,	PEPLCI,	PEMK,	PEMKSC,		
7	PEPMCI,	PETXFN,	PETXPR,	PECHXP,		
8	PECHSP,	PETXCI,	PECHH,	PECHUP,		
9	PETXP,	PETXAL,	PEATCH,	PEATCU,		
a	PEATP,	PEATAL,	PEANST,	PEIS,		
b	PEISI,	PEICI,	PEEDFG,	PEEDT,		
c	PEEWSC,	PEEDCI,	PEPA,	PEPRPV,		
d	PEPARF,	PEADS,	PERES,	PEIASF,		
e	PEHRID,	PELMT3,	PELMT,	PEGMT3,		
f	PEGMT,	PEMCV3,	PEMCV,	PEMCLI,		
g	PERMCV,	PEVWI,	PEEXST,	PELB,		
h	PEAP,	PEGSE,	PEPKID			
PARAMETER(	PEALL=0,	PENIL=1,	PEPL3=2,	PEPL=3,		
1	PEPM3=4,	PEPM=5,	PETX3=6,	PETX=7,		
2	PEATR3=8,	PEATR=9,	PEFA3=10,	PEFA=11,		
3	PEFAS3=12,	PEFAS=13,	PECA3=14,	PECA=15,		
4	PEGDP3=16,	PEGDP=17,	PEPLI=18,	PEPMI=19,		
5	PETXI=20,	PEII=21,	PEEDI=22,	PELN=23,		
6	PELWSC=24,	PEPLCI=25,	PEMK=26,	PEMKSC=27,		
7	PEPMCI=28,	PETXFN=29,	PETXPR=30,	PECHXP=31,		
8	PECHSP=32,	PETXCI=33,	PECHH=34,	PECHUP=35,		
9	PETXP=36,	PETXAL=37,	PEATCH=38,	PEATCU=39,		
a	PEATP=40,	PEATAL=41,	PEANST=42,	PEIS=43,		
b	PEISI=44,	PEICI=45,	PEEDFG=46,	PEEDT=47,		
c	PEEWSC=48,	PEEDCI=49,	PEPA=50,	PEPRPV=51,		
d	PEPARF=52,	PEADS=53,	PERES=54,	PEIASF=55,		
e	PEHRID=56,	PELMT3=57,	PELMT=58,	PEGMT3=59,		
f	PEGMT=60,	PEMCV3=61,	PEMCV=62,	PEMCLI=63,		
g	PERMCV=64,	PEVWI=65,	PEEXST=66,	PELB=67,		
h	PEAP=68,	PEGSE=69,	PEPKID=70			
GDP attributes	polyline	polymarker	text	interior	edge	
INTEGER	PPLATT,	PPMATT,	PTXATT,	PINATT,	PEDATT	
PARAMETER(	PPLATT=0,	PPMATT=1,	PTXATT=2,	PINATT=3,	PEDATT=4)	
input class	none,	locator,	stroke,	valuator,	choice,	pick,
	string					
INTEGER	PNCLAS,	PLOCAT,	PSTROK,	PVALUA,	PCHOIC,	PPICK,
1	PSTRIN					
PARAMETER(	PNCLAS=0,	PLOCAT=1,	PSTROK=2,	PVALUA=3,	PCHOIC=4,	PPICK=5,
1	PSTRIN=6)					
input device status	nonc,	ok,	nopick,	nochoicc		
INTEGER	PNONE,	POK,	PNPICK,	PNCHOI		
PARAMETER(	PNONE=0,	POK=1,	PNPICK=2,	PNCHOI=2)		
interior style	hollow,	solid,	pattern,	hatch,	empty	
INTEGER	PHOLLO,	PSOLID,	PPATTR,	PHATCH,	PISEMP	
PARAMETER(	PHOLLO=0,	PSOLID=1,	PPATTR=2,	PHATCH=3,	PISEMP=4)	

Enumeration types

linetype	solid, INTEGER PLSOLI, PARAMETER(PLSOLI=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	dash, PLDASH, PLDASH=2, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	dot, PLDOT, PLDOT=3, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	dash-dot PLDASD, PLDASD=4) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.
marker type	‘.’, INTEGER PPOINT, PARAMETER(PPOINT=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	‘+’, PPLUS, PPLUS=2, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	‘*’, PAST, PAST=3, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	‘o’, POMARK, POMARK=4, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.
modelling clipping operator	replace, INTEGER PMCREP, PARAMETER(PMCREP=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	intersect PMCINT, PMCINT=2) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
modification mode	NIVE, INTEGER PNIVE, PARAMETER(PNIVE=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	UWOR, PUWOR, PUWOR=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	UQUM PUQUM, PUQUM=2) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	
more simultaneous events	nomore, INTEGER PNMORE, PARAMETER(PNMORE=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	more PMORE, PMORE=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
off/on switch for edge flag and error handling mode	off, INTEGER POFF, PARAMETER(POFF=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	on PON, PON=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
open-structure status	none, INTEGER PNONST, PARAMETER(PNONST=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	open POPNST, POPNST=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
operating mode	request, INTEGER PREQU, PARAMETER(PREQU=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	sample, PSAMPL, PSAMPL=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	event PEVENT, PEVENT=2) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	
path order	topfirst, INTEGER PPOTOP, PARAMETER(PPOTOP=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	bottomfirst PPOBOT, PPOBOT=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
polyline/fill area control flag	polyline, INTEGER PPLINE, PARAMETER(PPLINE=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	fill area, PFILLA, PFILLA=1, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	fill area set PFILAS, PFILAS=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	
presence of invalid values	absent, INTEGER PABSNT, PARAMETER(PABSNT=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	present PPRSNT, PPRSNT=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
reference handling flag	delete, INTEGER PDELE, PARAMETER(PDELE=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	keep PKEEP, PKEEP=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		
regeneration flag	postpone, INTEGER PPOSTP, PARAMETER(PPOSTP=0, INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.	perform PPERFO, PPERFO=1) INTEGER rather than enumeration type. Explicitly defined and required portion of conceptually unbounded range defined here.		

Enumeration types

relative input priority	higher, INTEGER PHIGHR, PARAMETER(PHIGHR=0, PLOWER=1)	lower PLOWER				
search direction	backward, INTEGER PBWD, PARAMETER(PBWD=0, PFWD=1)	forward PFWD				
search success indicator	failure, INTEGER PFAIL, PARAMETER(PFAIL=0, PSUCC=1)	success PSUCC				
state of visual representation	correct, INTEGER PVROK, PARAMETER(PVROK=0, PVRDFR=1, PVRSIM=2)	deferred, PVRDFR	simulated PVRSIM			
structure network source	CSS, INTEGER PCSS, PARAMETER(PCSS=0, PARCHV=1)	ARCHIVE PARCHV				
structure state value	STCL, INTEGER PSTCL, PARAMETER(PSTCL=0, PSTOP=1)	STOP PSTOP				
structure status indicator	non-existent, INTEGER PSNOEX, PARAMETER(PSNOEX=0, PSEMP=1, PSNEMP=2)	empty, PSEMP	not empty PSNEMP			
system state value	PHCL, INTEGER PPHCL, PARAMETER(PPHCL=0, PPHOP=1)	PHOP PPHOP				
text alignment horizontal	normal, INTEGER PAHNOR, PARAMETER(PAHNOR=0, PALEFT=1, PACENT=2, PARITE=3)	left, PALEFT	centre, PACENT	right PARITE		
text alignment vertical	normal, INTEGER PAVNOR, PARAMETER(PAVNOR=0, PATOP=1, PACAP=2, PAHALF=3, PABASE=4, PABOTT=5)	top, PATOP	cap, PACAP	half, PAHALF	base, PABASE	bottom PABOTT
text path	right, INTEGER PRIGHT, PARAMETER(PRIGHT=0, PLEFT=1, PUP=2, PDOWN=3)	left, PLEFT	up, PUP	down PDOWN		
text precision	string, INTEGER PSTRP, PARAMETER(PSTRP=0, PCHARP=1, PSTRKP=2)	character, PCHARP	stroke PSTRKP			
type of returned values	set, INTEGER PSET, PARAMETER(PSET=0, PREALI=1)	realized PREALI				
update state	notpending, INTEGER PNPEND, PARAMETER(PNPEND=0, PPEND=1)	pending PPEND				
vector/raster/other type	vector, INTEGER PVECTR, PARAMETER(PVECTR=0, PRASTR=1, POTHWK=2)	raster, PRASTR	other POTHWK			

Enumeration types

viewtype	parallel, INTEGER PPARL, PARAMETER(PPARL=0,	perspective PPERS PPERS=1)			
workstation category	OUTPUT, INTEGER POUTPT, PARAMETER(POUTPT=0,	INPUT, PINPUT, PINPUT=1,	OUTIN, POUTIN, POUTIN=2,	MO, PMO, PMO=3,	MI PMI PMI=4)
workstation dependency indicator	workstation independent, INTEGER PWKI, PARAMETER(PWKI=0,	workstation dependent PWKD PWKD=1)			
workstation state value	WSCL, INTEGER PWSCL, PARAMETER(PWSCL=0,	WSOP PWSOP PWSOP=1)			
additional in the FORTRAN binding, to be used with PHIGS inquiry functions that return both Current and Requested values	current, INTEGER PCURVL, PARAMETER(PCURVL=0,	requested PRQSVL PRQSVL=1)			

PHIGS functions. These names are used for error handling. The names are the same as the PHIGS function names except that the sentinel character 'P' is replaced by 'E'. The same function identification is used for both full FORTRAN 77 and FORTRAN 77 Subset.

INTEGER	EOPPH,	ECLPH,	EOPWK,	ECLWK,	ERST
PARAMETER(	EOPPH=0,	ECLPH=1,	EOPWK=2,	ECLWK=3,	ERST=4)
INTEGER	EUWK,	ESDUS,	EMSG,	EPL3,	EPL
PARAMETER(	EUWK=5,	ESDUS=6,	EMSG=7,	EPL3=8,	EPL=9)
INTEGER	EPM3,	ETX3,	ETX3,	ETX,	EATR3
PARAMETER(	EPM3=10,	ETX3=11,	ETX3=12,	ETX=13,	EATR3=14)
INTEGER	EATR,	EFA3,	EFA,	EFAS3,	EFAS
PARAMETER(	EATR=15,	EFA3=16,	EFA=17,	EFAS3=18,	EFAS=19)
INTEGER	ECA3,	ECA,	EGDP3,	EGDP,	ESPLI
PARAMETER(	ECA3=20,	ECA=21,	EGDP3=22,	EGDP=23,	ESPLI=24)
INTEGER	ESPMI,	ESTXI,	ESII,	ESEDI,	ESLN
PARAMETER(	ESPMI=25,	ESTXI=26,	ESII=27,	ESEDI=28,	ESLN=29)
INTEGER	ESLWSC,	ESPLCI,	ESMK,	ESMKSC,	ESPMCI
PARAMETER(	ESLWSC=30,	ESPLCI=31,	ESMK=32,	ESMKSC=33,	ESPMCI=34)
INTEGER	ESTXFN,	ESTXPR,	ESCHXP,	ESCHSP,	ESTXCI
PARAMETER(	ESTXFN=35,	ESTXPR=36,	ESCHXP=37,	ESCHSP=38,	ESTXCI=39)
INTEGER	ESCHH,	ESCHUP,	ESTXP,	ESTXAL,	ESATCH
PARAMETER(	ESCHH=40,	ESCHUP=41,	ESTXP=42,	ESTXAL=43,	ESATCH=44)
INTEGER	ESATCU,	ESATP,	ESATAL,	ESANS,	ESIS
PARAMETER(	ESATCU=45,	ESATP=46,	ESATAL=47,	ESANS=48,	ESIS=49)
INTEGER	ESISI,	ESICI,	ESEDFG,	ESEDIT,	FSEWSC
PARAMETER(	ESISI=50,	ESICI=51,	ESEDFG=52,	ESEDIT=53,	ESEWSC=54)
INTEGER	ESEDCI,	ESPA,	ESPRPV,	ESPARF,	EADS
PARAMETER(	ESEDCI=55,	ESPA=56,	ESPRPV=57,	ESPARF=58,	EADS=59)
INTEGER	FRES,	ESIASF,	ESPLR,	ESPMR,	ESTXR
PARAMETER(	FRES=60,	ESIASF=61,	ESPLR=62,	ESPMR=63,	ESTXR=64)
INTEGER	ESIR,	ESEDR,	ESPAR,	ESCR,	ESHLFT
PARAMETER(	ESIR=65,	ESEDR=66,	ESPAR=67,	ESCR=68,	ESHLFT=69)
INTEGER	ESIVFT,	ESCMD,	ESHRID,	ESHRM,	ESLMT3
PARAMETER(	ESIVFT=70,	ESCMD=71,	ESHRID=72,	ESHRM=73,	ESLMT3=74)
INTEGER	ESLMT,	ESGMT3,	ESGMT,	ESMCV3,	ESMCV
PARAMETER(	ESLMT=75,	ESGMT3=76,	ESGMT=77,	ESMCV3=78,	ESMCV=79)
INTEGER	ESMCLI,	ERMCV,	ESVWI,	ESVWR3,	ESVWR
PARAMETER(	ESMCLI=80,	ERMCV=81,	ESVWI=82,	ESVWR3=83,	ESVWR=84)
INTEGER	ESWTKW3,	ESWKW,	ESWKV3,	ESWKV3,	ESWKV
PARAMETER(	ESWTKW3=85,	ESWKW=86,	ESWKV3=87,	ESWKV3=88,	ESWKV=89)
INTEGER	EOPST,	ECLST,	EEXSI,	ELB,	EAP
PARAMETER(	EOPST=90,	ECLST=91,	EEXST=92,	ELB=93,	EAP=94)
INTEGER	EGSE,	ESEDM,	ECELST,	ESEP,	EOSEP
PARAMETER(	EGSE=95,	ESEDM=96,	ECELST=97,	ESEP=98,	EOSEP=99)
INTEGER	ESEPLB,	EDEL,	EDELLA,	EDELLB,	EEMST
PARAMETER(	ESEPLB=100,	EDEL=101,	EDELLA=102,	EDELLB=103,	EEMST=104)
INTEGER	EDST,	EDSN,	EDAS,	ECSTID,	ECSTRF
PARAMETER(	EDST=105,	EDSN=106,	EDAS=107,	ECSTID=108,	ECSTRF=109)

Enumeration types

INTEGER	ECSTIR,	EPOST,	EUPOST,	EUPAST,	EOPARF
PARAMETER(	ECSTIR=110,	EPOST=111,	EUPOST=112,	EUPAST=113,	EOPARF=114)
INTEGER	ECLARF,	EARST,	EARSN,	EARAST,	ESCNRS
PARAMETER(	ECLARF=115,	EARST=116,	EARSN=117,	EARAST=118,	ESCNRS=119)
INTEGER	ERSID,	EREST,	ERESN,	ERAST,	EREPAN
PARAMETER(	ERSID=120,	EREST=121,	ERESN=122,	ERAST=123,	EREPAN=124)
INTEGER	EREPDE,	EDSTAR,	EDSNAR,	EDASAR,	ESPKID
PARAMETER(	EREPDE=125,	EDSTAR=126,	EDSNAR=127,	EDASAR=128,	ESPKID=129)
INTEGER	ESPKFT,	EINLC3,	EINLC,	EINSK3,	EINSK
PARAMETER(	ESPKFT=130,	EINLC3=131,	EINLC=132,	EINSK3=133,	EINSK=134)
INTEGER	EINV3,	EINVL,	EINCH3,	EINCH,	EINPK3
PARAMETER(	EINV3=135,	EINVL=136,	EINCH3=137,	EINCH=138,	EINPK3=139)
INTEGER	EINPK,	EINST3,	EINST,	ESLCM,	ESSKM
PARAMETER(	EINPK=140,	EINST3=141,	EINST=142,	ESLCM=143,	ESSKM=144)
INTEGER	ESVLM,	ESCHM,	ESPKM,	ESSTM,	ERQLC3
PARAMETER(	ESVLM=145,	ESCHM=146,	ESPKM=147,	ESSTM=148,	ERQLC3=149)
INTEGER	ERQLC,	ERQSK3,	ERQSK,	ERQVL,	ERQCH
PARAMETER(	ERQLC=150,	ERQSK3=151,	ERQSK=152,	ERQVL=153,	ERQCH=154)
INTEGER	ERQPK,	ERQST,	ESMLC3,	ESMLC,	ESMSK3
PARAMETER(	ERQPK=155,	ERQST=156,	ESMLC3=157,	ESMLC=158,	ESMSK3=159)
INTEGER	ESMSK,	ESMVL,	ESMCH,	ESMPK,	ESMST
PARAMETER(	ESMSK=160,	ESMVL=161,	ESMCH=162,	ESMPK=163,	ESMST=164)
INTEGER	EWAIT,	EFLUSH,	EGTLC3,	EGTLC,	EGTSK3
PARAMETER(	EWAIT=165,	EFLUSH=166,	EGTLC3=167,	EGTLC=168,	EGTSK3=169)
INTEGER	EGTSK,	EGTVL,	EGTCH,	EGTPK,	EGTST
PARAMETER(	EGTSK=170,	EGTVL=171,	EGTCH=172,	EGTPK=173,	EGTST=174)
INTEGER	EWITM,	EGTITM,	ERDITM,	EIITM,	ESERHM
PARAMETER(	EWITM=175,	EGTITM=176,	ERDITM=177,	EIITM=178,	ESERHM=179)
INTEGER	EESC,	EPREC,	EUREC		
PARAMETER(	EESC=180,	EPREC=181,	EUREC=182)		

7 List of the PHIGS function names

The complete list of PHIGS function names follows. PACK DATA RECORD and UNPACK DATA RECORD do not appear in ISO/IEC 9592-1, but have been created for this binding

7.1 List of functions ordered alphabetically by bound name

PADS	AD-S	ADD NAMES TO SET
PAP	AP	APPLICATION DATA
PARAST	ARA-ST	ARCHIVE ALL STRUCTURES
PARSN	AR-SN	ARCHIVE STRUCTURE NETWORKS
PARST	AR-ST	ARCHIVE STRUCTURES
PATR	ATR	ANNOTATION TEXT RELATIVE
PATR3	ATR-3	ANNOTATION TEXT RELATIVE 3
PATR3S	ATR-3-S	ANNOTATION TEXT RELATIVE 3 (FORTRAN 77 subset)
PATRS	ATR-S	ANNOTATION TEXT RELATIVE (FORTRAN 77 subset)
PBLTM	BL-T-M	BUILD TRANSFORMATION MATRIX
PBLTM3	BL-T-M-3	BUILD TRANSFORMATION MATRIX 3
PCA	C-A	CELL ARRAY
PCA3	C-A-3	CELL ARRAY 3
PCELST	C-EL-ST	COPY ALL ELEMENTS FROM STRUCTURE
PCLARF	CL-AR-F	CLOSE ARCHIVE FILE
PCLPH	CL-PH	CLOSE PHIGS
PCLST	CL-ST	CLOSE STRUCTURE
PCLWK	CL-WK	CLOSE WORKSTATION
PCOM	CO-M	COMPOSE MATRIX
PCOM3	CO-M-3	COMPOSE MATRIX 3
PCOTM	CO-T-M	COMPOSE TRANSFORMATION MATRIX
PCOTM3	CO-T-M-3	COMPOSE TRANSFORMATION MATRIX 3
PCSTID	CSTID	CHANGE STRUCTURE IDENTIFIER
PCSTIR	C-ST-IR	CHANGE STRUCTURE IDENTIFIER AND REFERENCES
PCSTRF	C-ST-RF	CHANGE STRUCTURE REFERENCES
PDAS	DAS	DELETE ALL STRUCTURES
PDASAR	DAS-AR	DELETE ALL STRUCTURES FROM ARCHIVE
PDEL	D-EL	DELETE ELEMENT
PDELLB	D-EL-LB	DELETE ELEMENTS BETWEEN LABELS
PDELRA	D-EL-RA	DELETE ELEMENT RANGE
PDSN	D-SN	DELETE STRUCTURE NETWORK
PDSNAR	D-SN-AR	DELETE STRUCTURE NETWORKS FROM ARCHIVE
PDST	D-ST	DELETE STRUCTURE
PDSTAR	D-ST-AR	DELETE STRUCTURES FROM ARCHIVE
PECLPH	E-CL-PH	EMERGENCY CLOSE PHIGS
PELS	EL-S	ELEMENT SEARCH
PEMST	EM-ST	EMPTY STRUCTURE
PERHND	ER-HND	ERROR HANDLING
PERLOG	ER-LOG	ERROR LOGGING
PESC	ESC	ESCAPE
PEVMM	EVMM	EVALUATE VIEW MAPPING MATRIX
PEVMM3	EVMM-3	EVALUATE VIEW MAPPING MATRIX 3
PEVOM	EVOM	EVALUATE VIEW ORIENTATION MATRIX
PEVOM3	EVOM-3	EVALUATE VIEW ORIENTATION MATRIX 3
PEXST	EX-ST	EXECUTE STRUCTURE
PFA	F-A	FILL AREA
PFA3	F-A-3	FILL AREA 3
PFAS	F-A-S	FILL AREA SET
PFAS3	F-A-S-3	FILL AREA SET 3
PFLUSH	FLUSH	FLUSH DEVICE EVENTS
PGDP	G-D-P	GENERALIZED DRAWING PRIMITIVE
PGDP3	G-D-P-3	GENERALIZED DRAWING PRIMITIVE 3
PGSE	GSE	GENERALIZED STRUCTURE ELEMENT

List of the PHIGS function names

List of functions ordered alphabetically by bound name

PGTCH	GT-CH	GET CHOICE
PGTITM	GT-ITM	GET ITEM TYPE FROM METAFILE
PGTLC	GT-LC	GET LOCATOR
PGTLC3	GT-LC-3	GET LOCATOR 3
PGTPK	GT-PK	GET PICK
PGTSK	GT-SK	GET STROKE
PGTSK3	GT-SK-3	GET STROKE 3
PGTST	GT-ST	GET STRING
PGTVL	GT-VL	GET VALUATOR
PIITM	I-ITM	INTERPRET ITEM
PINCH	IN-CH	INITIALIZE CHOICE
PINCH3	IN-CH-3	INITIALIZE CHOICE 3
PINLC	IN-LC	INITIALIZE LOCATOR
PINLC3	IN-LC-3	INITIALIZE LOCATOR 3
PINPK	IN-PK	INITIALIZE PICK
PINPK3	IN-PK-3	INITIALIZE PICK 3
PINSK	IN-SK	INITIALIZE STROKE
PINSK3	IN-SK-3	INITIALIZE STROKE 3
PINST	IN-ST	INITIALIZE STRING
PINST3	IN-ST-3	INITIALIZE STRING 3
PINVL	IN-VL	INITIALIZE VALUATOR
PINVL3	IN-VL-3	INITIALIZE VALUATOR 3
PISS	I-S-S	INCREMENTAL SPATIAL SEARCH
PISS3	I-S-S-3	INCREMENTAL SPATIAL SEARCH 3
PLB	LB	LABEL
PMSG	MSG	MESSAGE
PMSGS	MSG-S	MESSAGE (FORTRAN 77 subset)
POPARF	OP-AR-F	OPEN ARCHIVE FILE
POPPH	OP-PH	OPEN PHIGS
POPST	OP-ST	OPEN STRUCTURE
POPWK	OP-WK	OPEN WORKSTATION
POSEP	OS-EP	OFFSET ELEMENT POINTER
PPL	PL	POLYLINE
PPL3	PL-3	POLYLINE 3
PPM	PM	POLYMARKER
PPM3	PM-3	POLYMARKER 3
PPOST	PO-ST	POST STRUCTURE
PPREC	P-REC	PACK DATA RECORD
PQANF	Q-AN-F	INQUIRE ANNOTATION FACILITIES
PQARF	Q-AR-F	INQUIRE ARCHIVE FILES
PQARS	Q-AR-S	INQUIRE ARCHIVE STATE VALUE
PQCECO	Q-C-ECO	INQUIRE CURRENT ELEMENT CONTENT
PQCETS	Q-C-ETS	INQUIRE CURRENT ELEMENT TYPE AND SIZE
PQCF	Q-C-F	INQUIRE COLOUR FACILITIES
PQCHS	Q-CH-S	INQUIRE CHOICE DEVICE STATE
PQCHS3	Q-CH-S-3	INQUIRE CHOICE DEVICE STATE 3
PQCMD	Q-C-MD	INQUIRE COLOUR MODEL
PQCMDF	Q-C-MD-F	INQUIRE COLOUR MODEL FACILITIES
PQCNRS	Q-CN-RS	INQUIRE CONFLICT RESOLUTION
PQCR	Q-C-R	INQUIRE COLOUR REPRESENTATION
PQCST	Q-C-ST	INQUIRE ALL CONFLICTING STRUCTURES
PQCSTN	Q-C-ST-N	INQUIRE CONFLICTING STRUCTURES IN NETWORK
PQDCH	Q-D-CH	INQUIRE DEFAULT CHOICE DEVICE DATA
PQDCH3	Q-D-CH-3	INQUIRE DEFAULT CHOICE DEVICE DATA 3
PQDDUS	Q-D-D-U-S	INQUIRE DEFAULT DISPLAY UPDATE STATE
PQDLC	Q-D-LC	INQUIRE DEFAULT LOCATOR DEVICE DATA
PQDLC3	Q-D-LC-3	INQUIRE DEFAULT LOCATOR DEVICE DATA 3
PQDP	Q-D-P	INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED
PQDPK	Q-D-PK	INQUIRE DEFAULT PICK DEVICE DATA
PQDPK3	Q-D-PK-3	INQUIRE DEFAULT PICK DEVICE DATA 3
PQDSK	Q-D-SK	INQUIRE DEFAULT STROKE DEVICE DATA
PQDSK3	Q-D-SK-3	INQUIRE DEFAULT STROKE DEVICE DATA 3

List of functions ordered alphabetically by bound name

List of the PHIGS function names

PQDSP	Q-D-SP	INQUIRE DISPLAY SPACE SIZE
PQDSP3	Q-D-SP-3	INQUIRE DISPLAY SPACE SIZE 3
PQDST	Q-D-ST	INQUIRE DEFAULT STRING DEVICE DATA
PQDST3	Q-D-ST-3	INQUIRE DEFAULT STRING DEVICE DATA 3
PQDSTR	Q-DSTR	INQUIRE DYNAMICS OF STRUCTURES
PQDSWA	Q-DSWA	INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES
PQDUS	Q-D-U-S	INQUIRE DISPLAY UPDATE STATE
PQDVL	Q-D-VL	INQUIRE DEFAULT VALUATOR DEVICE DATA
PQDVL3	Q-D-VL-3	INQUIRE DEFAULT VALUATOR DEVICE DATA 3
PQECI	Q-E-C-I	INQUIRE LIST OF COLOUR INDICES
PQECO	Q-ECO	INQUIRE ELEMENT CONTENT
PQEDF	Q-ED-F	INQUIRE EDGE FACILITIES
PQEDM	Q-ED-M	INQUIRE EDIT MODE
PQEDR	Q-ED-R	INQUIRE EDGE REPRESENTATION
PQEEDI	Q-E-ED-I	INQUIRE LIST OF EDGE INDICES
PQEGD3	Q-EGD3	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3
PQEGDP	Q-E-G-D-P	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES
PQEGSE	Q-E-GSE	INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS
PQEII	Q-E-I-I	INQUIRE LIST OF INTERIOR INDICES
PQEP	Q-EP	INQUIRE ELEMENT POINTER
PQEPAI	Q-E-PA-I	INQUIRE LIST OF PATTERN INDICES
PQEPLI	Q-E-PL-I	INQUIRE LIST OF POLYLINE INDICES
PQEPMI	Q-E-PM-I	INQUIRE LIST OF POLYMARKER INDICES
PQERHM	Q-ERHM	INQUIRE ERROR HANDLING MODE
PQETS	Q-ETS	INQUIRE ELEMENT TYPE AND SIZE
PQETXI	Q-E-TX-I	INQUIRE LIST OF TEXT INDICES
PQEVWI	Q-E-VW-I	INQUIRE LIST OF VIEW INDICES
PQEWK	Q-E-WK	INQUIRE LIST OF AVAILABLE WORKSTATION TYPES
PQGDP	Q-G-D-P	INQUIRE GENERALIZED DRAWING PRIMITIVE
PQGDP3	Q-G-D-P-3	INQUIRE GENERALIZED DRAWING PRIMITIVE 3
PQGSEF	Q-GSE-F	INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES
PQHLFT	Q-HL-FT	INQUIRE HIGHLIGHTING FILTER
PQHRIF	Q-HR-I-F	INQUIRE HLHSR (IDENTIFIER) FACILITIES
PQHRM	Q-HR-M	INQUIRE HLHSR MODE
PQHRMF	Q-HR-M-F	INQUIRE HLHSR (MODE) FACILITIES
PQIF	Q-I-F	INQUIRE INTERIOR FACILITIES
PQIQOV	Q-I-Q-OV	INQUIRE INPUT QUEUE OVERFLOW
PQIR	Q-I-R	INQUIRE INTERIOR REPRESENTATION
PQIVFT	Q-IV-FT	INQUIRE INVISIBILITY FILTER
PQLCS	Q-LC-S	INQUIRE LOCATOR DEVICE STATE
PQLCS3	Q-LC-S-3	INQUIRE LOCATOR DEVICE STATE 3
PQLI	Q-L-I	INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES
PQMCLF	Q-MCL-F	INQUIRE MODELLING CLIPPING FACILITIES
PQOPST	Q-OP-ST	INQUIRE OPEN STRUCTURE
PQOPWK	Q-OP-WK	INQUIRE SET OF OPEN WORKSTATIONS
PQPAF	Q-PA-F	INQUIRE PATTERN FACILITIES
PQPAN	Q-P-AN	INQUIRE PATHS TO ANCESTORS
PQPAR	Q-PA-R	INQUIRE PATTERN REPRESENTATION
PQPCR	Q-P-C-R	INQUIRE PREDEFINED COLOUR REPRESENTATION
PQPDE	Q-P-DE	INQUIRE PATHS TO DESCENDANTS
PQPEDR	Q-P-ED-R	INQUIRE PREDEFINED EDGE REPRESENTATION
PQPHF	Q-PH-F	INQUIRE PHIGS FACILITIES
PQPIR	Q-P-I-R	INQUIRE PREDEFINED INTERIOR REPRESENTATION
POPKS	Q-PK-S	INQUIRE PICK DEVICE STATE
PQPKS3	Q-PK-S-3	INQUIRE PICK DEVICE STATE 3
PQPLF	Q-PL-F	INQUIRE POLYLINE FACILITIES
PQPLR	Q-PL-R	INQUIRE POLYLINE REPRESENTATION
PQPMF	Q-PM-F	INQUIRE POLYMARKER FACILITIES
PQPMR	Q-PM-R	INQUIRE POLYMARKER REPRESENTATION
PQPOST	Q-PO-ST	INQUIRE POSTED STRUCTURES
PQPPAR	Q-P-PA-R	INQUIRE PREDEFINED PATTERN REPRESENTATION
PQPLR	Q-P-PL-R	INQUIRE PREDEFINED POLYLINE REPRESENTATION

List of the PHIGS function names

List of functions ordered alphabetically by bound name

PQPPMR	Q-P-PM-R	INQUIRE PREDEFINED POLYMARKER REPRESENTATION
PQPTXR	Q-P-TX-R	INQUIRE PREDEFINED TEXT REPRESENTATION
PQPVWR	Q-P-VW-R	INQUIRE PREDEFINED VIEW REPRESENTATION
PQSID	Q-SID	INQUIRE STRUCTURE IDENTIFIERS
PQSIM	Q-SIM	INQUIRE MORE SIMULTANEOUS EVENTS
PQSKS	Q-SK-S	INQUIRE STROKE DEVICE STATE
PQSKS3	Q-SK-S-3	INQUIRE STROKE DEVICE STATE 3
PQSTRS	Q-STRS	INQUIRE STRUCTURE STATE VALUE
POSTS	Q-ST-S	INQUIRE STRING DEVICE STATE
POSTS3	Q-ST-S-3	INQUIRE STRING DEVICE STATE 3
PQSTST	Q-ST-ST	INQUIRE STRUCTURE STATUS
PQSYS	Q-SY-S	INQUIRE SYSTEM STATE VALUE
POTXF	Q-TX-F	INQUIRE TEXT FACILITIES
PQTXR	Q-TX-R	INQUIRE TEXT REPRESENTATION
PQTXS	Q-TX-X	INQUIRE TEXT EXTENT
PQTXXS	Q-TX-X-S	INQUIRE TEXT EXTENT (FORTRAN 77 subset)
PQVLS	Q-VL-S	INQUIRE VALUATOR DEVICE STATE
PQVLS3	Q-VL-S-3	INQUIRE VALUATOR DEVICE STATE 3
PQVWF	Q-VW-F	INQUIRE VIEW FACILITIES
PQVWR	Q-VW-R	INQUIRE VIEW REPRESENTATION
PQWKC	Q-WK-C	INQUIRE WORKSTATION CONNECTION AND TYPE
PQWKCA	Q-WK-CA	INQUIRE WORKSTATION CATEGORY
PQWKCL	Q-WK-CL	INQUIRE WORKSTATION CLASSIFICATION
PQWKPO	Q-WK-PO	INQUIRE SET OF WORKSTATIONS TO WHICH POSTED
PQWKS	Q-WK-S-L	INQUIRE WORKSTATION STATE TABLE LENGTHS
PQWKST	Q-WKST	INQUIRE WORKSTATION STATE VALUE
PQWKT	Q-WK-T	INQUIRE WORKSTATION TRANSFORMATION
PQWKT3	Q-WK-T-3	INQUIRE WORKSTATION TRANSFORMATION 3
PRAST	RA-ST	RETRIEVE ALL STRUCTURES
PRDITM	RD-ITM	READ ITEM FROM METAFILE
PREPAN	RE-P-AN	RETRIEVE PATHS TO ANCESTORS
PREPDE	RE-P-DE	RETRIEVE PATHS TO DESCENDANTS
PRES	RE-S	REMOVE NAMES FROM SET
PRESN	RE-SN	RETRIEVE STRUCTURE NETWORKS
PREST	RE-ST	RETRIEVE STRUCTURES
PRMCV	R-MCV	RESTORE MODELLING CLIPPING VOLUME
PRO	RO	ROTATE
PROX	RO-X	ROTATE X
PROY	RO-Y	ROTATE Y
PROZ	RO-Z	ROTATE Z
PROCH	RQ-CH	REQUEST CHOICE
PRQLC	RQ-LC	REQUEST LOCATOR
PRQLC3	RQ-LC-3	REQUEST LOCATOR 3
PRQPK	RQ-PK	REQUEST PICK
PRQSK	RQ-SK	REQUEST STROKE
PRQSK3	RQ-SK-3	REQUEST STROKE 3
PRQST	RQ-ST	REQUEST STRING
PRQVL	RQ-VL	REQUEST VALUATOR
PRSID	RSID	RETRIEVE STRUCTURE IDENTIFIERS
PRST	R-ST	REDRAW ALL STRUCTURES
PSANS	S-AN-S	SET ANNOTATION STYLE
PSATAL	SAT-AL	SET ANNOTATION TEXT ALIGNMENT
PSATCH	S-ATCH	SET ANNOTATION TEXT CHARACTER HEIGHT
PSATCU	S-ATCU	SET ANNOTATION TEXT CHARACTER UP VECTOR
PSATP	SAT-P	SET ANNOTATION TEXT PATH
PSC	SC	SCALE
PSC3	SC-3	SCALE 3
PSCHH	S-CH-H	SET CHARACTER HEIGHT
PSCHM	S-CH-M	SET CHOICE MODE
PSCHSP	S-CH-SP	SET CHARACTER SPACING
PSCHUP	S-CH-UP	SET CHARACTER UP VECTOR
PSCHXP	S-CH-XP	SET CHARACTER EXPANSION FACTOR

List of functions ordered alphabetically by bound name

List of the PHIGS function names

PSCMD	S-C-MD	SET COLOUR MODEL
PSCNRS	S-CN-RS	SET CONFLICT RESOLUTION
PSCR	S-C-R	SET COLOUR REPRESENTATION
PSDUS	S-D-U-S	SET DISPLAY UPDATE STATE
PSEDCI	S-EDCI	SET EDGE COLOUR INDEX
PSEDFG	S-EDFG	SET EDGE FLAG
PSEDI	S-ED-I	SET EDGE INDEX
PSEDM	S-ED-M	SET EDIT MODE
PSEDR	S-ED-R	SET EDGE REPRESENTATION
PSEDT	S-EDT	SET EDGETYPE
PSEP	S-EP	SET ELEMENT POINTER
PSEPLB	S-EP-LB	SET ELEMENT POINTER AT LABEL
PSERHM	S-ERHM	SET ERROR HANDLING MODE
PSEWSC	S-EW-SC	SET EDGEWIDTH SCALE FACTOR
PSGMT	S-GM-T	SET GLOBAL TRANSFORMATION
PSGMT3	S-GM-T-3	SET GLOBAL TRANSFORMATION 3
PSHLFT	S-HL-FT	SET HIGHLIGHTING FILTER
PSHRID	S-HR-ID	SET HLHSR IDENTIFIER
PSHRM	S-HR-M	SET HLHSR MODE
PSIASF	S-IASF	SET INDIVIDUAL ASF
PSICI	S-I-C-I	SET INTERIOR COLOUR INDEX
PSII	S-I-I	SET INTERIOR INDEX
PSIR	S-I-R	SET INTERIOR REPRESENTATION
PSIS	S-I-S	SET INTERIOR STYLE
PSISI	S-I-S-I	SET INTERIOR STYLE INDEX
PSIVFT	S-IV-FT	SET INVISIBILITY FILTER
PSLCM	S-LC-M	SET LOCATOR MODE
PSLMT	S-LM-T	SET LOCAL TRANSFORMATION
PSLMT3	S-LM-T-3	SET LOCAL TRANSFORMATION 3
PSLN	S-LN	SET LINETYPE
PSLWSC	S-LW-SC	SET LINEWIDTH SCALE FACTOR
PSMCH	SM-CH	SAMPLE CHOICE
PSMCLI	S-MCL-I	SET MODELLING CLIPPING INDICATOR
PSMCV	S-MCV	SET MODELLING CLIPPING VOLUME
PSMCV3	S-MCV-3	SET MODELLING CLIPPING VOLUME 3
PSMK	S-MK	SET MARKER TYPE
PSMKSC	S-MK-SC	SET MARKER SIZE SCALE FACTOR
PSMLC	SM-LC	SAMPLE LOCATOR
PSMLC3	SM-LC-3	SAMPLE LOCATOR 3
PSMPK	SM-PK	SAMPLE PICK
PSMSK	SM-SK	SAMPLE STROKE
PSMSK3	SM-SK-3	SAMPLE STROKE 3
PSMST	SM-ST	SAMPLE STRING
PSMVL	SM-VL	SAMPLE VALUATOR
PSPA	S-PA	SET PATTERN SIZE
PSPAR	S-PA-R	SET PATTERN REPRESENTATION
PSPARF	S-PARF	SET PATTERN REFERENCE POINT
PSPKFT	S-PK-FT	SET PICK FILTER
PSPKID	S-PK-ID	SET PICK IDENTIFIER
PSPKM	S-PK-M	SET PICK MODE
PSPLCI	S-PL-C-I	SET POLYLINE COLOUR INDEX
PSPLI	S-PL-I	SET POLYLINE INDEX
PSPLR	S-PL-R	SET POLYLINE REPRESENTATION
PSPMCI	S-PM-C-I	SET POLYMARKER COLOUR INDEX
PSPMI	S-PM-I	SET POLYMARKER INDEX
PSPMR	S-PM-R	SET POLYMARKER REPRESENTATION
PSPRPV	S-PRPV	SET PATTERN REFERENCE POINT AND VECTORS
PSSKM	S-SK-M	SET STROKE MODE
PSSTM	S-ST-M	SET STRING MODE
PSTXAL	S-TX-AL	SET TEXT ALIGNMENT
PSTXCI	S-TX-C-I	SET TEXT COLOUR INDEX
PSTXFN	S-TXFN	SET TEXT FONT

List of the PHIGS function names**List of functions ordered alphabetically by bound name**

PSTXI	S-TX-I	SET TEXT INDEX
PSTXP	S-TX-P	SET TEXT PATH
PSTXPR	S-TX-PR	SET TEXT PRECISION
PSTXR	S-TX-R	SET TEXT REPRESENTATION
PSVLM	S-VL-M	SET VALUATOR MODE
PSVTIP	S-VT-I-P	SET VIEW TRANSFORMATION INPUT PRIORITY
PSVWI	S-VW-I	SET VIEW INDEX
PSVWR	S-VW-R	SET VIEW REPRESENTATION
PSVWR3	S-VW-R-3	SET VIEW REPRESENTATION 3
PSWKV	S-WK-V	SET WORKSTATION VIEWPORT
PSWKV3	S-WK-V-3	SET WORKSTATION VIEWPORT 3
PSWKW	S-WK-W	SET WORKSTATION WINDOW
PSWKW3	S-WK-W-3	SET WORKSTATION WINDOW 3
PTP	TP	TRANSFORM POINT
PTP3	TP-3	TRANSFORM POINT 3
PTR	TR	TRANSLATE
PTR3	TR-3	TRANSLATE 3
PTX	TX	TEXT
PTX3	TX-3	TEXT 3
PTX3S	TX-3-S	TEXT 3 (FORTRAN 77 subset)
PTXS	TX-S	TEXT (FORTRAN 77 subset)
PUPAST	UPA-ST	UNPOST ALL STRUCTURES
PUPOST	UPO-ST	UNPOST STRUCTURE
PUREC	U-REC	UNPACK DATA RECORD
PUWK	U-WK	UPDATE WORKSTATION
PWAIT	WAIT	AWAIT EVENT
PWITM	W-ITM	WRITE ITEM TO METAFILE

7.2 List of functions ordered alphabetically by PHIGS function name

PADS	AD-S	ADD NAMES TO SET
PATR	ATR	ANNOTATION TEXT RELATIVE
PATRS	ATR-S	ANNOTATION TEXT RELATIVE (FORTRAN 77 subset)
PATR3	ATR-3	ANNOTATION TEXT RELATIVE 3
PATR3S	ATR-3-S	ANNOTATION TEXT RELATIVE 3 (FORTRAN 77 subset)
PAP	AP	APPLICATION DATA
PARAST	ARA-ST	ARCHIVE ALL STRUCTURES
PARSN	AR-SN	ARCHIVE STRUCTURE NETWORKS
PARST	AR-ST	ARCHIVE STRUCTURES
PWAIT	WAIT	AWAIT EVENT
PBLTM	BL-T-M	BUILD TRANSFORMATION MATRIX
PBLTM3	BL-T-M-3	BUILD TRANSFORMATION MATRIX 3
PCA	C-A	CELL ARRAY
PCA3	C-A-3	CELL ARRAY 3
PCSTID	CSTID	CHANGE STRUCTURE IDENTIFIER
PCSTIR	C-ST-IR	CHANGE STRUCTURE IDENTIFIER AND REFERENCES
PCSTRF	C-ST-RF	CHANGE STRUCTURE REFERENCES
PCLARF	CL-AR-F	CLOSE ARCHIVE FILE
PCLPH	CL-PH	CLOSE PHIGS
PCLST	CL-ST	CLOSE STRUCTURE
PCLWK	CL-WK	CLOSE WORKSTATION
PCOM	CO-M	COMPOSE MATRIX
PCOM3	CO-M-3	COMPOSE MATRIX 3
PCOTM	CO-T-M	COMPOSE TRANSFORMATION MATRIX
PCOTM3	CO-T-M-3	COMPOSE TRANSFORMATION MATRIX 3
PCELST	C-EL-ST	COPY ALL ELEMENTS FROM STRUCTURE
PDAS	DAS	DELETE ALL STRUCTURES
PDASAR	DAS-AR	DELETE ALL STRUCTURES FROM ARCHIVE
PDEL	D-EL	DELETE ELEMENT
PDELRA	D-EL-RA	DELETE ELEMENT RANGE
PDELLB	D-EL-LB	DELETE ELEMENTS BETWEEN LABELS
PDST	D-ST	DELETE STRUCTURE
PDSN	D-SN	DELETE STRUCTURE NETWORK

List of functions ordered alphabetically by PHIGS function name List of the PHIGS function names

PDSNAR	D-SN-AR	DELETE STRUCTURE NETWORKS FROM ARCHIVE
PDSTAR	D-ST-AR	DELETE STRUCTURES FROM ARCHIVE
PELS	EL-S	ELEMENT SEARCH
PECLPH	E-CL-PH	EMERGENCY CLOSE PHIGS
PEMST	EM-ST	EMPTY STRUCTURE
PERHND	ER-HND	ERROR HANDLING
PERLOG	ER-LOG	ERROR LOGGING
PESC	ESC	ESCAPE
PEVMM	EVMM	EVALUATE VIEW MAPPING MATRIX
PEVMM3	EVMM-3	EVALUATE VIEW MAPPING MATRIX 3
PEVOM	EVOM	EVALUATE VIEW ORIENTATION MATRIX
PEVOM3	EVOM-3	EVALUATE VIEW ORIENTATION MATRIX 3
PEXST	EX-ST	EXECUTE STRUCTURE
PFA	F-A	FILL AREA
PFA3	F-A-3	FILL AREA 3
PFAS	F-A-S	FILL AREA SET
PFAS3	F-A-S-3	FILL AREA SET 3
PFLUSH	FLUSH	FLUSH DEVICE EVENTS
PGDP	G-D-P	GENERALIZED DRAWING PRIMITIVE
PGDP3	G-D-P-3	GENERALIZED DRAWING PRIMITIVE 3
PGSE	GSE	GENERALIZED STRUCTURE ELEMENT
PGTCH	GT-CH	GET CHOICE
PGTITM	GT-ITM	GET ITEM TYPE FROM METAFILE
PGTLC	GT-LC	GET LOCATOR
PGTLC3	GT-LC-3	GET LOCATOR 3
PGTPK	GT-PK	GET PICK
PGTST	GT-ST	GET STRING
PGTSK	GT-SK	GET STROKE
PGTSK3	GT-SK-3	GET STROKE 3
PGTVL	GT-VL	GET VALUATOR
PISS	I-S-S	INCREMENTAL SPATIAL SEARCH
PISS3	I-S-S-3	INCREMENTAL SPATIAL SEARCH 3
PINCH	IN-CH	INITIALIZE CHOICE
PINCH3	IN-CH-3	INITIALIZE CHOICE 3
PINLC	IN-LC	INITIALIZE LOCATOR
PINLC3	IN-LC-3	INITIALIZE LOCATOR 3
PINPK	IN-PK	INITIALIZE PICK
PINPK3	IN-PK-3	INITIALIZE PICK 3
PINST	IN-ST	INITIALIZE STRING
PINST3	IN-ST-3	INITIALIZE STRING 3
PINSK	IN-SK	INITIALIZE STROKE
PINSK3	IN-SK-3	INITIALIZE STROKE 3
PINVL	IN-VL	INITIALIZE VALUATOR
PINVL3	IN-VL-3	INITIALIZE VALUATOR 3
PQCST	Q-C-ST	INQUIRE ALL CONFLICTING STRUCTURES
PQANF	Q-AN-F	INQUIRE ANNOTATION FACILITIES
PQARF	Q-AR-F	INQUIRE ARCHIVE FILES
PQARS	Q-AR-S	INQUIRE ARCHIVE STATE VALUE
PQCHS	Q-CH-S	INQUIRE CHOICE DEVICE STATE
PQCHS3	Q-CH-S-3	INQUIRE CHOICE DEVICE STATE 3
PQCF	Q-C-F	INQUIRE COLOUR FACILITIES
PQCMD	Q-C-MD	INQUIRE COLOUR MODEL
PQCMDF	Q-C-MD-F	INQUIRE COLOUR MODEL FACILITIES
PQCR	Q-C-R	INQUIRE COLOUR REPRESENTATION
PQCNRS	Q-CN-RS	INQUIRE CONFLICT RESOLUTION
PQCSTN	Q-C-ST-N	INQUIRE CONFLICTING STRUCTURES IN NETWORK
PQCECO	Q-C-ECO	INQUIRE CURRENT ELEMENT CONTENT
PQCETS	Q-C-ETS	INQUIRE CURRENT ELEMENT TYPE AND SIZE
PQDCH	Q-D-CH	INQUIRE DEFAULT CHOICE DEVICE DATA
PQDCH3	Q-D-CH-3	INQUIRE DEFAULT CHOICE DEVICE DATA 3
PQDDUS	Q-D-D-U-S	INQUIRE DEFAULT DISPLAY UPDATE STATE
PQDLC	Q-D-LC	INQUIRE DEFAULT LOCATOR DEVICE DATA

List of the PHIGS function names List of functions ordered alphabetically by PHIGS function name

PQDLC3	Q-D-LC-3	INQUIRE DEFAULT LOCATOR DEVICE DATA 3
PQDPK	Q-D-PK	INQUIRE DEFAULT PICK DEVICE DATA
PQDPK3	Q-D-PK-3	INQUIRE DEFAULT PICK DEVICE DATA 3
PQDST	Q-D-ST	INQUIRE DEFAULT STRING DEVICE DATA
PQDST3	Q-D-ST-3	INQUIRE DEFAULT STRING DEVICE DATA 3
PQDSK	Q-D-SK	INQUIRE DEFAULT STROKE DEVICE DATA
PQDSK3	Q-D-SK-3	INQUIRE DEFAULT STROKE DEVICE DATA 3
PQDVL	Q-D-VL	INQUIRE DEFAULT VALUATOR DEVICE DATA
PQDVL3	Q-D-VL-3	INQUIRE DEFAULT VALUATOR DEVICE DATA 3
PQDSP	Q-D-SP	INQUIRE DISPLAY SPACE SIZE
PQDSP3	Q-D-SP-3	INQUIRE DISPLAY SPACE SIZE 3
PQDUS	Q-D-U-S	INQUIRE DISPLAY UPDATE STATE
PQDSTR	Q-DSTR	INQUIRE DYNAMICS OF STRUCTURES
PQDSWA	Q-DSWA	INQUIRE DYNAMICS OF WORKSTATION ATTRIBUTES
PQEDF	Q-ED-F	INQUIRE EDGE FACILITIES
PQEDR	Q-ED-R	INQUIRE EDGE REPRESENTATION
PQEDM	Q-ED-M	INQUIRE EDIT MODE
PQECO	Q-ECO	INQUIRE ELEMENT CONTENT
PQEP	Q-EP	INQUIRE ELEMENT POINTER
PQETS	Q-ETS	INQUIRE ELEMENT TYPE AND SIZE
PQERHM	Q-ERHM	INQUIRE ERROR HANDLING MODE
PQGDP	Q-G-D-P	INQUIRE GENERALIZED DRAWING PRIMITIVE
PQGDP3	Q-G-D-P-3	INQUIRE GENERALIZED DRAWING PRIMITIVE 3
PQGSEF	Q-GSE-F	INQUIRE GENERALIZED STRUCTURE ELEMENT FACILITIES
PQHLFT	Q-HL-FT	INQUIRE HIGHLIGHTING FILTER
PQHRIF	Q-HR-I-F	INQUIRE HLHSR (IDENTIFIER) FACILITIES
PQHRMF	Q-HR-M-F	INQUIRE HLHSR (MODE) FACILITIES
PQHRM	Q-HR-M	INQUIRE HLHSR MODE
POIOOV	Q-I-O-OV	INQUIRE INPUT QUEUE OVERFLOW
PQIF	Q-I-F	INQUIRE INTERIOR FACILITIES
PQIR	Q-I-R	INQUIRE INTERIOR REPRESENTATION
PQIVFT	Q-IV-FT	INQUIRE INVISIBILITY FILTER
PQEGDP	Q-E-G-D-P	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES
PQEGD3	Q-EGD3	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3
PQEGSE	Q-E-GSE	INQUIRE LIST OF AVAILABLE GENERALIZED STRUCTURE ELEMENTS
PQEWK	Q-E-WK	INQUIRE LIST OF AVAILABLE WORKSTATION TYPES
PQECI	Q-E-C-I	INQUIRE LIST OF COLOUR INDICES
PQEEDI	Q-E-ED-I	INQUIRE LIST OF EDGE INDICES
PQEII	Q-E-I-I	INQUIRE LIST OF INTERIOR INDICES
PQEPAI	Q-E-PA-I	INQUIRE LIST OF PATTERN INDICES
PQEPLI	Q-E-PL-I	INQUIRE LIST OF POLYLINE INDICES
PQEPMI	Q-E-PM-I	INQUIRE LIST OF POLYMARKER INDICES
PQETXI	Q-E-TX-I	INQUIRE LIST OF TEXT INDICES
PQEVWI	Q-E-VW-I	INQUIRE LIST OF VIEW INDICES
PQLCS	Q-LC-S	INQUIRE LOCATOR DEVICE STATE
PQLCS3	Q-LC-S-3	INQUIRE LOCATOR DEVICE STATE 3
POMCLF	Q-MCL-F	INQUIRE MODELLING CLIPPING FACILITIES
PQSIM	Q-SIM	INQUIRE MORE SIMULTANEOUS EVENTS
PQLI	Q-L-I	INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES
PQDP	Q-D-P	INQUIRE NUMBER OF DISPLAY PRIORITIES SUPPORTED
PQOPST	Q-OP-ST	INQUIRE OPEN STRUCTURE
POPAN	Q-P-AN	INQUIRE PATHS TO ANCESTORS
POPDE	Q-P-DE	INQUIRE PATHS TO DESCENDANTS
POPAF	Q-PA-F	INQUIRE PATTERN FACILITIES
POPAPR	Q-PA-R	INQUIRE PATTERN REPRESENTATION
POPHEF	Q-PH-F	INQUIRE PHIGS FACILITIES
POPKS	Q-PK-S	INQUIRE PICK DEVICE STATE
POPKS3	Q-PK-S-3	INQUIRE PICK DEVICE STATE 3
PQPLF	Q-PL-F	INQUIRE POLYLINE FACILITIES
PQPLR	Q-PL-R	INQUIRE POLYLINE REPRESENTATION
PQPMF	Q-PM-F	INQUIRE POLYMARKER FACILITIES
PQPMR	Q-PM-R	INQUIRE POLYMARKER REPRESENTATION

List of functions ordered alphabetically by PHIGS function name

List of the PHIGS function names

PQPOST	Q-PO-ST	INQUIRE POSTED STRUCTURES
PQPCR	Q-P-C-R	INQUIRE PREDEFINED COLOUR REPRESENTATION
PQPEDR	Q-P-ED-R	INQUIRE PREDEFINED EDGE REPRESENTATION
PQPIR	Q-P-I-R	INQUIRE PREDEFINED INTERIOR REPRESENTATION
PQPPAR	Q-P-PA-R	INQUIRE PREDEFINED PATTERN REPRESENTATION
PQPPLR	Q-P-PL-R	INQUIRE PREDEFINED POLYLINE REPRESENTATION
POPMPR	Q-P-PM-R	INQUIRE PREDEFINED POLYMARKER REPRESENTATION
PQPTXR	Q-P-TX-R	INQUIRE PREDEFINED TEXT REPRESENTATION
PQPVWR	Q-P-VW-R	INQUIRE PREDEFINED VIEW REPRESENTATION
PQOPWK	Q-OP-WK	INQUIRE SET OF OPEN WORKSTATIONS
PQWKPO	Q-WK-PO	INQUIRE SET OF WORKSTATIONS TO WHICH POSTED
POSTS	Q-ST-S	INQUIRE STRING DEVICE STATE
PQSTS3	Q-ST-S-3	INQUIRE STRING DEVICE STATE 3
PQSKS	Q-SK-S	INQUIRE STROKE DEVICE STATE
PQSKS3	Q-SK-S-3	INQUIRE STROKE DEVICE STATE 3
PQSID	Q-SID	INQUIRE STRUCTURE IDENTIFIERS
PQSTRS	Q-STRS	INQUIRE STRUCTURE STATE VALUE
PQSTST	Q-ST-ST	INQUIRE STRUCTURE STATUS
PQSYS	Q-SY-S	INQUIRE SYSTEM STATE VALUE
PQTXX	Q-TX-X	INQUIRE TEXT EXTENT
PQTXXS	Q-TX-X-S	INQUIRE TEXT EXTENT (FORTRAN 77 subset)
PQTXF	Q-TX-F	INQUIRE TEXT FACILITIES
PQTXR	Q-TX-R	INQUIRE TEXT REPRESENTATION
PQVLS	Q-VL-S	INQUIRE VALUATOR DEVICE STATE
PQVLS3	Q-VL-S-3	INQUIRE VALUATOR DEVICE STATE 3
PQVWF	Q-VW-F	INQUIRE VIEW FACILITIES
PQVWR	Q-VW-R	INQUIRE VIEW REPRESENTATION
PQWKCA	Q-WK-CA	INQUIRE WORKSTATION CATEGORY
PQWKCL	Q-WK-CL	INQUIRE WORKSTATION CLASSIFICATION
POWKC	Q-WK-C	INQUIRE WORKSTATION CONNECTION AND TYPE
PQWKS	Q-WK-S-L	INQUIRE WORKSTATION STATE TABLE LENGTHS
PQWKST	Q-WKST	INQUIRE WORKSTATION STATE VALUE
PQWKT	Q-WK-T	INQUIRE WORKSTATION TRANSFORMATION
PQWKT3	Q-WK-T-3	INQUIRE WORKSTATION TRANSFORMATION 3
PIITM	I-ITM	INTERPRET ITEM
PLB	LB	LABEL
PMSG	MSG	MESSAGE
PMSGS	MSG-S	MESSAGE (FORTRAN 77 subset)
POSEP	OS-EP	OFFSET ELEMENT POINTER
POPAPF	OP-AR-F	OPEN ARCHIVE FILE
POPPH	OP-PH	OPEN PHIGS
POPST	OP-ST	OPEN STRUCTURE
POPWK	OP-WK	OPEN WORKSTATION
PPREC	P-REC	PACK DATA RECORD
PPL	PL	POLYLINE
PPL3	PL-3	POLYLINE 3
PPM	PM	POLYMARKER
PPM3	PM-3	POLYMARKER 3
PPOST	PO-ST	POST STRUCTURE
PRDITM	RD-ITM	READ ITEM FROM METAFILE
PRST	R-ST	REDRAW ALL STRUCTURES
PRES	RE-S	REMOVE NAMES FROM SET
PROCH	RQ-CH	REQUEST CHOICE
PRQLC	RQ-LC	REQUEST LOCATOR
PRQLC3	RQ-LC-3	REQUEST LOCATOR 3
PROPK	RQ-PK	REQUEST PICK
PROST	RQ-ST	REQUEST STRING
PROSK	RQ-SK	REQUEST STROKE
PROSK3	RQ-SK-3	REQUEST STROKE 3
PROVL	RQ-VL	REQUEST VALUATOR
PRMCV	R-MCV	RESTORE MODELLING CLIPPING VOLUME
PRAST	RA-ST	RETRIEVE ALL STRUCTURES

List of the PHIGS function names List of functions ordered alphabetically by PHIGS function name

PREPAN	RE-P-AN	RETRIEVE PATHS TO ANCESTORS
PREPDE	RE-P-DE	RETRIEVE PATHS TO DESCENDANTS
PRSID	RSID	RETRIEVE STRUCTURE IDENTIFIERS
PRESN	RE-SN	RETRIEVE STRUCTURE NETWORKS
PREST	RE-ST	RETRIEVE STRUCTURES
PRO	RO	ROTATE
PROX	RO-X	ROTATE X
PROY	RO-Y	ROTATE Y
PROZ	RO-Z	ROTATE Z
PSMCH	SM-CH	SAMPLE CHOICE
PSMLC	SM-LC	SAMPLE LOCATOR
PSMLC3	SM-LC-3	SAMPLE LOCATOR 3
PSMPK	SM-PK	SAMPLE PICK
PSMST	SM-ST	SAMPLE STRING
PSMSK	SM-SK	SAMPLE STROKE
PSMSK3	SM-SK-3	SAMPLE STROKE 3
PSMVL	SM-VL	SAMPLE VALUATOR
PSC	SC	SCALE
PSC3	SC-3	SCALE 3
PSANS	S-AN-S	SET ANNOTATION STYLE
PSATAL	SAT-AL	SET ANNOTATION TEXT ALIGNMENT
PSATCH	S-ATCH	SET ANNOTATION TEXT CHARACTER HEIGHT
PSATCU	S-ATCU	SET ANNOTATION TEXT CHARACTER UP VECTOR
PSATP	SAT-P	SET ANNOTATION TEXT PATH
PSCHXP	S-CH-XP	SET CHARACTER EXPANSION FACTOR
PSCHH	S-CH-H	SET CHARACTER HEIGHT
PSCHSP	S-CH-SP	SET CHARACTER SPACING
PSCHUP	S-CH-UP	SET CHARACTER UP VECTOR
PSCHM	S-CH-M	SET CHOICE MODE
PSCMD	S-C-MD	SET COLOUR MODEL
PSCR	S-C-R	SET COLOUR REPRESENTATION
PSCNRS	S-CN-RS	SET CONFLICT RESOLUTION
PSDUS	S-D-U-S	SET DISPLAY UPDATE STATE
PSEDCI	S-EDCI	SET EDGE COLOUR INDEX
PSEDFG	S-EDFG	SET EDGE FLAG
PSEDI	S-ED-I	SET EDGE INDEX
PSEDR	S-ED-R	SET EDGE REPRESENTATION
PSEDT	S-EDT	SET EDGETYPE
PSEWSC	S-EW-SC	SET EDGEWIDTH SCALE FACTOR
PSEDM	S-ED-M	SET EDIT MODE
PSEP	S-EP	SET ELEMENT POINTER
PSEPLB	S-EP-LB	SET ELEMENT POINTER AT LABEL
PSERHM	S-ERHM	SET ERROR HANDLING MODE
PSGMT	S-GM-T	SET GLOBAL TRANSFORMATION
PSGMT3	S-GM-T-3	SET GLOBAL TRANSFORMATION 3
PSHLFT	S-HL-FT	SET HIGHLIGHTING FILTER
PSHRID	S-HR-ID	SET HLHSR IDENTIFIER
PSHRM	S-HR-M	SET HLHSR MODE
PSIASF	S-IASF	SET INDIVIDUAL ASF
PSICI	S-I-C-I	SET INTERIOR COLOUR INDEX
PSII	S-I-I	SET INTERIOR INDEX
PSIR	S-I-R	SET INTERIOR REPRESENTATION
PSIS	S-I-S	SET INTERIOR STYLE
PSISI	S-I-S-I	SET INTERIOR STYLE INDEX
PSIVFT	S-IV-FT	SET INVISIBILITY FILTER
PSLN	S-LN	SET LINETYPE
PSLWSC	S-LW-SC	SET LINEWIDTH SCALE FACTOR
PSLMT	S-LM-T	SET LOCAL TRANSFORMATION
PSLMT3	S-LM-T-3	SET LOCAL TRANSFORMATION 3
PSLCM	S-LC-M	SET LOCATOR MODE
PSMKSC	S-MK-SC	SET MARKER SIZE SCALE FACTOR
PSMK	S-MK	SET MARKER TYPE

List of functions ordered alphabetically by PHIGS function name List of the PHIGS function names

PSMCLI	S-MCL-I	SET MODELLING CLIPPING INDICATOR
PSMCV	S-MCV	SET MODELLING CLIPPING VOLUME
PSMCV3	S-MCV-3	SET MODELLING CLIPPING VOLUME 3
PSPARF	S-PARF	SET PATTERN REFERENCE POINT
PSPRPV	S-PRPV	SET PATTERN REFERENCE POINT AND VECTORS
PSPAR	S-PA-R	SET PATTERN REPRESENTATION
PSPA	S-PA	SET PATTERN SIZE
PSPKFT	S-PK-FT	SET PICK FILTER
PSPKID	S-PK-ID	SET PICK IDENTIFIER
PSPKM	S-PK-M	SET PICK MODE
PSPLCI	S-PL-C-I	SET POLYLINE COLOUR INDEX
PSPLI	S-PL-I	SET POLYLINE INDEX
PSPLR	S-PL-R	SET POLYLINE REPRESENTATION
PSPMCI	S-PM-C-I	SET POLYMARKER COLOUR INDEX
PSPMI	S-PM-I	SET POLYMARKER INDEX
PSPMR	S-PM-R	SET POLYMARKER REPRESENTATION
PSSTM	S-ST-M	SET STRING MODE
PSSKM	S-SK-M	SET STROKE MODE
PSTXAL	S-TX-AL	SET TEXT ALIGNMENT
PSTXCI	S-TX-C-I	SET TEXT COLOUR INDEX
PSTXFN	S-TXFN	SET TEXT FONT
PSTXI	S-TX-I	SET TEXT INDEX
PSTXP	S-TX-P	SET TEXT PATH
PSTXPR	S-TX-PR	SET TEXT PRECISION
PSTXR	S-TX-R	SET TEXT REPRESENTATION
PSVLM	S-VL-M	SET VALUATOR MODE
PSVWI	S-VW-I	SET VIEW INDEX
PSVWR	S-VW-R	SET VIEW REPRESENTATION
PSVWR3	S-VW-R-3	SET VIEW REPRESENTATION 3
PSVTIP	S-VT-I-P	SET VIEW TRANSFORMATION INPUT PRIORITY
PSWKV	S-WK-V	SET WORKSTATION VIEWPORT
PSWKV3	S-WK-V-3	SET WORKSTATION VIEWPORT 3
PSWKW	S-WK-W	SET WORKSTATION WINDOW
PSWKW3	S-WK-W-3	SET WORKSTATION WINDOW 3
PTX	TX	TEXT
PTXS	TX-S	TEXT (FORTRAN 77 subset)
PTX3	TX-3	TEXT 3
PTX3S	TX-3-S	TEXT 3 (FORTRAN 77 subset)
PTP	TP	TRANSFORM POINT
PTP3	TP-3	TRANSFORM POINT 3
PTR	TR	TRANSLATE
PTR3	TR-3	TRANSLATE 3
PUREC	U-REC	UNPACK DATA RECORD
PUPAST	UPA-ST	UNPOST ALL STRUCTURES
PUPOST	UPO-ST	UNPOST STRUCTURE
PUWK	U-WK	UPDATE WORKSTATION
PWITM	W-ITM	WRITE ITEM TO METAFILE

8 PHIGS errors specific to the FORTRAN binding

Certain features of the FORTRAN language make additional errors (beyond the ones described in ISO/IEC 9592-1) possible. Specifically, these new errors are defined:

- 2000 Ignoring function, enumeration type out of range -- the INTEGER passed as a PHIGS enumerated type is not within the range of valid values.
- 2001 Ignoring function, output parameter size insufficient -- a FORTRAN array or string being passed as an output parameter is too small to contain the returned information.
- 2002 Ignoring function, list or set element not available -- for a non-empty list or set, a value less than zero or greater than the size of a list or set was passed as the requested list or set element in an inquiry routine.
- 2003 Ignoring function, invalid data record -- the data record passed to a PHIGS routine cannot be decoded, or there was a problem encountered when PHIGS was creating a data record, making the result invalid.
- 2004 Ignoring function, input parameter size out of range -- the INTEGER passed as an input parameter defining the size of FORTRAN array parameters is negative or greater than an absolute maximum (e.g. for FORTRAN subset functions taking strings as input parameters).
- 2005 Ignoring function, invalid list of point lists -- one of the conditions defining a valid list of point list has been violated:
 - a) $NPL \geq 1$,
 - b) PXA, PYA and PZA are dimensioned by at least $IXA(NPL)$ (however that it is allowable for the implementation not to generate this error in this case),
 - c) $IXA(1) \geq 3$ (the first point list is at least 3 points),
 - d) $IXA(i+1) - IXA(i) \geq 3$ for $i=1$ to $NPL-1$, when $NPL \geq 2$ (the i th point list is at least 3 points).
- 2006 Ignoring function, invalid list of filters -- one of the conditions defining a valid list of filters has been violated:
 - e) $FLN \geq 1$,
 - f) FLIS and FLES are dimensioned by at least $FLESX(FLN)$ (however that it is allowable for the implementation not to generate this error in this case).

9 The PHIGS function interface

9.1 General principles

For each PHIGS function the corresponding FORTRAN SUBROUTINE declaration is given. The name of the PHIGS function is listed, followed by its FORTRAN name and the corresponding parameters. After that, the list of parameters is described by type and a brief identifying phrase.

For the mapping of ENUMERATION types see clause 6.

For GENERALIZED DRAWING PRIMITIVE, GENERALIZED DRAWING PRIMITIVE 3, GENERALIZED STRUCTURE ELEMENT, and ESCAPE, subroutines PGDP, PGDP3, PGSE, and PESC are defined. Each GDP with an identifier less than 10,000 in the ISO International Register of Graphical Items and that is available in the implementation may, in addition, be accessed by a subroutine of the form PD*pqrs* or P3*pqrs*, where *pqrs* is a string of 4 digits representing the GDP identifier with leading zeros if necessary.¹⁾ The parameters are derived from those of PGDP as follows: N, PXA, PYA (and PZA for 3D) are required; PRIMID is absent; the data record is specified by parameters appropriate to the particular GDP identifier. Similarly, each ESCAPE and GSE with an identifier less than 10,000 in the ISO International Register of Graphical Items and that is available in the implementation may, in addition, be accessed by a subroutine of the form PE*pqrs* (for ESCAPE) and PS*pqrs* (for GSE). The parameters are derived from those of PESC and PGSE as follows: FCTID is absent and the data record is specified by parameters appropriate to the specific ESCAPE and GSE function. The type and order of the parameters representing the data record for the individual subroutines, and the corresponding data sent to PACK DATA RECORD if PGDP, PGDP3, PGSE or PESC is used, are defined in the ISO International Register of Graphical Items. For GDP, GDP3, GSE and ESCAPE, it is possible to define each separate subroutine using PGDP, PGDP3, PGSE and PESC.

Unregistered GDP, GDP3, GSE, and ESCAPE functions may be accessed either by the general subroutines PGDP, PGDP3, PGSE, and PESC with negative identifiers, or by subroutines named PU*lmn*, PU3*lmn*, PUS*lmn*, and PUE*lmn* where *lmn* is the absolute value of the negative identifiers. Parameter lists for these subroutines are implementation dependent.

¹⁾ For the purpose of this International Standard and according to the rules for the designation and operation of registration authorities in the ISO Directives, the ISO Council has designated the National Institute of Standards and Technology (Institute of Computer Sciences and Technology), A-266 Technology Building, Gaithersburg, MD 20899, USA to act as registration authority.

The PHIGS function interface

Control functions

9.2 Control functions

OPEN PHIGS

SUBROUTINE POPPH (ERRFIL,BUFA)

Input Parameters:

INTEGER ERRFIL

error file

INTEGER BUFA

amount of memory units (implementation dependent; if -1, use implementation dependent default)

CLOSE PHIGS

SUBROUTINE PCLPH

OPEN WORKSTATION

SUBROUTINE POPWK (WKID,CONID,WTYPE)

Input Parameters:

INTEGER WKID

workstation identifier

INTEGER CONID

connection identifier

INTEGER WTYPE

workstation type

CLOSE WORKSTATION

SUBROUTINE PCLWK (WKID)

Input Parameters:

INTEGER WKID

workstation identifier

REDRAW ALL STRUCTURES

SUBROUTINE PRST (WKID,COFL)

Input Parameters:

INTEGER WKID

workstation identifier

INTEGER COFL

control flag (PCONDI, PALWAY)

UPDATE WORKSTATION

SUBROUTINE PUWK (WKID,REGFL)

Input Parameters:

INTEGER WKID

workstation identifier

INTEGER REGFL

regeneration flag (PPOSTP,PPERFO)

SET DISPLAY UPDATE STATE

SUBROUTINE PSDUS (WKID,DEFMOD,MODMOD)

Input Parameters:

INTEGER WKID

workstation identifier

INTEGER DEFMOD

deferral mode

(PASAP,PBNIG,PBNIL,PASTI,PWAITD)

INTEGER MODMOD

modification mode

(PNIVE,PUWOR,PUQUM)

MESSAGE

Full FORTRAN 77 version

SUBROUTINE PMSG (WKID,MESS)

Input Parameters:

INTEGER WKID

workstation identifier

CHARACTER*(*) MESS

message

MESSAGE

FORTAN 77 subset version

SUBROUTINE PMSGS (WKID,LSTR,MESS)

Input Parameters:

INTEGER WKID

workstation identifier

INTEGER LSTR

length of string (in characters)

CHARACTER*80 MESS

message

9.3 Output primitive functions

Output primitive functions

The PHIGS function interface

TEXT 3

FORTRAN 77 subset version

SUBROUTINE PTX3S (PX,PY,PZ,TDX,TDY,TDZ,LSTR,CHARS)

Input Parameters:

REAL PX, PY, PZ	text position (MC)
REAL TDX(2), TDY(2), TDZ(2)	text direction vectors (MC)
INTEGER LSTR	length of string (in characters)
CHARACTER*80 CHARS	character string

TEXT

Full FORTRAN 77 version

SUBROUTINE PTX (PX,PY,CHARS)

Input Parameters:

REAL PX, PY	text position (MC)
CHARACTER*(*) CHARS	character string

TEXT

FORTRAN 77 subset version

SUBROUTINE PTXS (PX,PY,LSTR,CHARS)

Input Parameters:

REAL PX, PY	text position (MC)
INTEGER LSTR	length of string (in characters)
CHARACTER*80 CHARS	character string

ANNOTATION TEXT RELATIVE 3

Full FORTRAN 77 version

SUBROUTINE PATR3 (RPX,RPY,RPZ,APX,APY,APZ,CHARS)

Input Parameters:

REAL RPX, RPY, RPZ	reference point (MC)
REAL APX, APY, APZ	annotation offset (NPC)
CHARACTER*(*) CHARS	character string

The PHIGS function interface**Output primitive functions****ANNOTATION TEXT RELATIVE 3**

FORTRAN 77 subset version

SUBROUTINE PATR3S (RPX,RPY,RPZ,APX,APY,APZ,LSTR,CHARS)

Input Parameters:

REAL RPX, RPY, RPZ

reference point (MC)

REAL APX, APY, APZ

annotation offset (NPC)

INTEGER LSTR

length of string (in characters)

CHARACTER*80 CHARS

character string

ANNOTATION TEXT RELATIVE

Full FORTRAN 77 version

SUBROUTINE PATR (RPX,RPY,APX,APY,CHARS)

Input Parameters:

REAL RPX, RPY

reference point (MC)

REAL APX, APY

annotation offset (NPC)

CHARACTER*(*) CHARS

character string

ANNOTATION TEXT RELATIVE

FORTRAN 77 subset version

SUBROUTINE PATRS (RPX,RPY,APX,APY,LSTR,CHARS)

Input Parameters:

REAL RPX, RPY

reference point (MC)

REAL APX, APY

annotation offset (NPC)

INTEGER LSTR

length of string (in characters)

CHARACTER*80 CHARS

character string

FILL AREA 3

SUBROUTINE PFA3 (N,PXA,PYA,PZA)

Input Parameters:

INTEGER N

number of points

REAL PXA (N), PYA (N), PZA (N) coordinates of points (MC)

FILL AREA

SUBROUTINE PFA (N,PXA,PYA)

Input Parameters:

INTEGER N

number of points

REAL PXA (N), PYA (N)

coordinates of points (MC)

FILL AREA SET 3

SUBROUTINE PFAS3 (NPL,IXA,PXA,PYA,PZA)

Input Parameters:

INTEGER NPL

number of point lists

INTEGER IXA(NPL)

array of end indices for the point lists

REAL PXA (*), PYA (*), PZA (*)

coordinates of points (MC) (see clause 5)

FILL AREA SET

SUBROUTINE PFAS (NPL,IXA,PXA,PYA)

Input Parameters:

INTEGER NPL

number of point lists

INTEGER IXA(NPL)

array of end indices for the point lists

REAL PXA (*), PYA (*)

coordinates of points (MC) (see clause 5)

CELL ARRAY 3

SUBROUTINE PCA3 (CPXA,CPYA,CPZA,DIMX,DIMY,ISC,ISR,DX,DY,COLIA)

Input Parameters:

REAL CPXA(3),CPYA(3),CPZA(3)

cell parallelogram (P,Q,R) (MC)

INTEGER DIMX, DIMY

the dimensions of COLIA which contains the cell array

INTEGER ISC, ISR

indices of start column, start row

INTEGER DX, DY

number of columns, number of rows

INTEGER COLIA (DIMX,DIMY)

colour index array

The PHIGS function interface

Output primitive functions

CELL ARRAY

SUBROUTINE PCA (PX,PY,QX,QY,DIMX,DIMY,ISC,ISR,DX,DY,COLIA)

Input Parameters:

REAL PX, PY, QX, QY	two points (P,Q) (MC)
INTEGER DIMX, DIMY	the dimensions of COLIA which contains the cell array
INTEGER ISC, ISR	indices of start column, start row
INTEGER DX, DY	number of columns, number of rows
INTEGER COLIA (DIMX,DIMY)	colour index array

GENERALIZED DRAWING PRIMITIVE 3

SUBROUTINE PGDP3 (N,PXA,PYA,PZA,PRIMID,LDR,DATREC)

Input Parameters:

INTEGER N	number of points (≥ 0)
REAL PXA (*), PYA (*), PZA (*)	coordinates of points (MC)
INTEGER PRIMID	GDP 3 identifier
INTEGER LDR	dimension of data record array
CHARACTER*80 DATREC(LDR)	data record

GENERALIZED DRAWING PRIMITIVE

SUBROUTINE PGDP (N,PXA,PYA,PRIMID,LDR,DATREC)

Input Parameters:

INTEGER N	number of points (≥ 0)
REAL PXA (*), PYA (*)	coordinates of points (MC)
INTEGER PRIMID	GDP identifier
INTEGER LDR	dimension of data record array
CHARACTER*80 DATREC(LDR)	data record

9.4 Attribute specification functions

9.4.1 Bundled attribute selection

SET POLYLINE INDEX

SUBROUTINE PSPLI (PLI)

Input Parameters:
INTEGER PLI polyline index

SET POLYMARKER INDEX

SUBROUTINE PSPMI (PMI)

Input Parameters:
INTEGER PMI polymarker index

SET TEXT INDEX

SUBROUTINE PSTXI (TXI)

Input Parameters:
INTEGER TXI text index

SET INTERIOR INDEX

SUBROUTINE PSII (II)

Input Parameters:
INTEGER II interior index

SET EDGE INDEX

SUBROUTINE PSEDI (EDI)

Input Parameters:
INTEGER EDI edge index