INTERNATIONAL STANDARD

ISO/IEC 9592-1

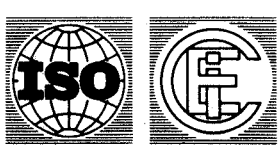
First edition
1989-04-01

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

Part 1 :
Functional description

Systèmes de traitement de l'information — Infographie — Interface de programmation du système graphique hiérarchisé (PHIGS) —

Partie 1 : Description fonctionnelle



Reference number
ISO/IEC 9592-1:1989 (E)

Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Contents

Page

0 Introduction 1
1 Scope and field of application 3
2 References 4
3 Definitions 5
4 The Programmer's Hierarchical Interactive Graphics System 15
4.1 About this part of ISO/IEC 9592 15
4.1.1 Specification and conformance 15
4.1.2 Registration 15
4.2 Overview 16
4.3 Concepts 18
4.3.1 PHIGS concepts 18
4.3.2 Relationship to ISO 7942 (GKS) and ISO 8805 (GKS-3D) 20
4.3.3 Notational conventions 20
4.4 The centralized structure store 22
4.4.1 Structure elements and structures 22
4.4.2 Structure networks 24
4.4.3 Structure traversal and display 25
4.4.4 Structure editing 28
4.4.5 Manipulation of structures in CSS 29
4.4.6 CSS search and inquiry 30
4.4.7 Structure archival and retrieval 32
4.4.8 Generalized Structure Elements (GSE) 33
4.4.9 Application data 33
4.5 Graphical output 34
4.5.1 Structure elements and output primitives 34
4.5.2 Output primitive attributes 38
4.5.3 Polyline attributes 43
4.5.4 Polymarker attributes 43
4.5.5 Text attributes 44
4.5.6 Annotation text attributes 55
4.5.7 Text extent and concatenation 55
4.5.8 Fill area attributes 58
4.5.9 Fill area set attributes 62
4.5.10 Cell array attributes 63
4.5.11 Generalized drawing primitive attributes 63
4.5.12 Colour 63
4.5.13 View index 64
4.5.14 Hidden line / hidden surface removal (HLHSR) identifier 64

© ISO/IEC 1989
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

Documents
Document Sharing Hub
Full document details: ISO_IEC 9592-1_1989

4.5.15 Name set attribute 64
4.5.16 Minimal simulations 65
4.5.17 Degenerate primitives 65
4.6 Workstations 67
4.6.1 Workstation characteristics 67
4.6.2 Workstation selection 68
4.6.3 Controlling picture changes 68
4.6.4 Clearing the display surface 75
4.6.5 Sending messages to a workstation 75
4.6.6 Hidden line / hidden surface removal 76
4.7 Coordinate systems and transformations 77
4.7.1 Coordinate system handedness 77
4.7.2 Modelling transformations and clipping 77
4.7.3 Modelling utility functions 79
4.7.4 Viewing 80
4.7.5 Viewing utility functions 84
4.7.6 Workstation transformation 90
4.7.7 Transformation of locator input 92
4.7.8 Transformation of stroke input 93
4.8 Graphical input 95
4.8.1 Introduction to logical input devices 95
4.8.2 Logical input device model 96
4.8.3 Operating modes of logical input devices 97
4.8.4 Measures of each input class 100
4.8.5 Input queue and current event report 101
4.8.6 Initialization of input devices 102
4.8.7 Locator and stroke input using 2D input 104
4.9 PHIGS metafile interface 105
4.10 PHIGS states 107
4.11 Inquiry functions 108
4.12 Error handling 109
4.13 Special interfaces between PHIGS and application program 112
4.14 Minimum support criteria 113
5 PHIGS Functional Specification 116
5.1 Notational conventions 116
5.2 Control functions 117
5.3 Output primitive functions 122
5.4 Attribute specification functions 132
5.4.1 Bundled attribute selection 132
5.4.2 Individual attribute selection 133
5.4.3 Aspect source flag setting 150
5.4.4 Workstation attribute table definition 150
5.4.5 Workstation filter definition 158
5.4.6 Colour model control 159
5.4.7 HLHSR attributes 160
5.5 Transformation and clipping functions 161
5.5.1 Modelling transformations and clipping 161
5.5.2 View operations 165
5.5.3 Workstation transformation 167
5.5.4 Utility functions to support modelling 169
5.5.5 Utility functions to support viewing 176
5.6 Structure content functions 179
5.7 Structure manipulation functions 185
5.8 Structure display functions 188
5.9 Structure archiving functions 190
5.10 Input functions 197
5.10.1 Pick identifier and filter 197
5.10.2 Initialization of input devices 197

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

5.10.3 Setting the mode of input devices 215
5.10.4 Request input functions 218
5.10.5 Sample input functions 221
5.10.6 Event input functions 225
5.11 Metafile functions 230
5.12 Inquiry functions 232
5.12.1 Introduction 232
5.12.2 Inquiry functions for operating state values 232
5.12.3 Inquiry functions for PHIGS description table 233
5.12.4 Inquiry functions for PHIGS state list 235
5.12.5 Inquiry functions for workstation state list 238
5.12.6 Inquiry functions for workstation description table 259
5.12.7 Inquiry function for structure state list 288
5.12.8 Inquiry functions for structure content 289
5.12.9 Inquiry functions for error state list 300
5.13 Error control functions 302
5.14 Special interface function 304
6 PHIGS data structures 305
6.1 Notation and data types 305
6.2 Operating states 308
6.3 PHIGS description table 309
6.4 PHIGS traversal state list 312
6.5 PHIGS state list 314
6.6 Workstation state list 316
6.7 Workstation description table 320
6.8 Structure state list 326
6.9 PHIGS error state list 327

Annexes

A Function Lists 328
A.1 Alphabetic 328
A.2 Order of appearance 334
B Error list 341
B.1 Implementation dependent 341
B.2 States 341
B.3 Workstations 341
B.4 Output attributes 341
B.5 Transformations and viewing 342
B.6 Structures 342
B.7 Input 343
B.8 Metafiles 343
B.9 Escape 343
B.10 Archive / retrieve 343
B.11 Miscellaneous 344
B.12 System 344
B.13 Reserved errors 344
C Interfaces 345
C.1 Introduction 345
C.2 Language Binding 345
C.3 Implementation 346
D Allowable differences in PHIGS implementations 348
D.1 Introduction 348
D.2 Global differences 348
D.3 Workstation dependent differences 349
E The PHIGS viewing model 352
F PHIGS/GKS differences 353

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

G HLHSR considerations 355
H Relationship of CGM and PHIGS 356
H.1 Introduction 356
H.2 Scope 356
H.3 Overview of the differences between PHIGS and CGM 356
H.4 Mapping concepts 357
H.4.1 Principles 357
H.4.2 Workstations 357
H.4.3 Picture generation 358
H.4.4 Picture input 358
H.4.5 Coordinates and clipping 359
H.4.6 Workstation transformation 359
H.4.7 Colour table 360
H.4.8 Set representation 360
H.5 Metafile generation 360
H.5.1 Control functions 360
H.5.2 Structure traversal 362
H.5.3 Metafile description 363
H.5.4 User items 364
H.6 Interpretation of CGM by PHIGS 364
H.7 Mapping between item types and elements 366
I Colour models 367
I.1 Introduction 367
I.2 RGB colour model 368
I.3 CIELUV colour model 368
I.3.1 CIE XYZ colour space 368
I.3.2 CIE 1931 (Y,x,y) space 369
I.3.3 The CIE 1976 (L*u*v*) CIELUV uniform colour space 372
I.3.4 Colour differences 373
I.4 HSV colour model 374
I.5 HLS colour model 375
I.6 Conversion between colour models 375
I.6.1 CIE XYZ reference model 375
I.6.2 Conversion between CIELUV and CIE XYZ models 376
I.6.3 Conversion between RGB and CIE XYZ models 376
I.6.3.1 Derivation of conversion factors 376
I.6.3.2 Conversion from RGB to CIE XYZ 377
I.6.3.3 Conversion from CIE XYZ to RGB 377
I.6.3.4 Representation of black 377
I.6.3.5 Example conversion 377

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form a system for worldwide standardization as a whole. 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 approval before their acceptance as International Standards. They are approved in accordance with procedures requiring at least 75 % approval by the national bodies voting.

International Standard ISO/IEC 9592-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

ISO/IEC 9592 consists of the following parts, under the general title *Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS)*:

— *Part 1: Functional description*

— *Part 2: Archive file format*

— *Part 3: Clear-text encoding of archive file*

Annex D forms an integral part of this part of ISO/IEC 9592. Annexes A, B, C, E, F, G, H, I are for information only.

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

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

Part 1 :
Functional description

0 Introduction

The Programmer's Hierarchical Interactive Graphics System (PHIGS) provides a set of functions for

- definition, display and modification of 2D or 3D graphical data,
- definition, display and manipulation of geometrically related objects,
- modification of graphics data and the relationships between the graphical data.

This International Standard draws extensively on GKS (Graphical Kernel System ISO 7942) and GKS-3D (Graphical Kernel System for Three Dimensions ISO 8805) for its model and functionality. In addition this International Standard enables graphical (and application) data to be stored in a hierarchical data store. Information in the data store can be inserted, modified and deleted with the provided functions. The relationship of this part of ISO/IEC 9592 to GKS and GKS-3D is further described in 4.3.2.

The choice of which graphics standard to use will depend on a number of factors: application profile, overall system architecture, equipment available, existing application database interaction, system performance considerations, user interface requirements, management policy and other external factors. The aim of producing a compatible set of graphics standards in GKS, GKS-3D and PHIGS is to allow that choice to be made in the most flexible way.

The main reasons for introducing a standard in this area of computer graphics are

a) to allow application programs using dynamic hierarchical graphics to be easily portable between installations,

b) to aid the understanding and use of dynamic hierarchical graphics methods by application programmers;

c) to reduce program development costs and time; many of the functions currently performed by the application program will now be performed by PHIGS;

d) to serve manufacturers of graphics equipment as a guideline in providing useful combinations of graphics capabilities in a device.

To meet these objectives, a number of design principles were adopted:

Documents
Document Sharing Hub
Full document details: ISO_IEC 9592-1_1989

e) Consistency: the mandatory requirements of PHIGS should not be mutually contradictive.

f) Compatibility: this Standard will be compatible with GKS and GKS-3D except when technical reasons justify differences.

g) Orthogonality: the functions should be independent of each other.

h) Completeness: all the functions necessary for application programs to use a dynamic hierarchical graphics system should be included.

i) Minimality: redundant functions are only supported where their availability enables application programs to improve performance or where some collection of capabilities is frequently used.

j) Programmer Experience: those using PHIGS should have a working knowledge of computer graphics.

k) Error Handling: error conditions should be minimized, and their impact well defined.

l) Device Independence: PHIGS should allow an application program to address facilities of different graphics input and output devices with minimal changes to the application program.

m) Device Dependence: PHIGS should allow an application program to address specific graphics input and output devices in a direct manner.

n) Implementability: it should be possible to support PHIGS functions using most languages on most operating systems.

o) Efficiency: PHIGS should be capable of being implemented and executed without consuming undue amounts of computer resources.

p) Interaction: Some application programs will require realtime or near-realtime response from PHIGS. PHIGS will not exclude such application programs though specific graphics devices and dedicated computer resources may be necessary.

Annexes A to C and E to I are given for information; they do not form part of this part of ISO/IEC 9592.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

1 Scope and field of application

This part of ISO/IEC 9592 specifies a set of functions for computer graphics programming, the Programmer's Hierarchical Interactive Graphics System (PHIGS). PHIGS is a graphics system for application programs that produce computer generated pictures on line graphics or raster graphics output devices. It supports operator input and interactions by supplying basic functions for graphical input and hierarchical picture definition. Picture definitions are retained in a *centralized structure store* where they may be edited by an application.

Pictures are displayed on *workstations* consisting of a single output device and a number of input devices. Several workstations can be used simultaneously. The application program is allowed to adapt its behaviour at a workstation to make best use of workstation capabilities.

Functions are specified for archiving picture definitions to file. In addition an interface to the Computer Graphics Metafile (ISO 8632) is described.

NOTE - For certain parameters of the functions, PHIGS defines value ranges as being reserved for registration (see 4.1.2). The meanings of these values will be defined using the established procedures.

This part of ISO/IEC 9592 defines a language independent nucleus of a graphics system for integration into a programming language. PHIGS is embedded in a language layer obeying the particular conventions of the language. Such language bindings are specified for ISO or ISO/IEC languages in ISO/IEC 9593.

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

2 References

ISO 646, *Information processing - ISO 7-bit coded character set for information interchange.*

ISO 2022, *Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.*

ISO 2382-13, *Data processing - Vocabulary - Part 13: Computer graphics.*

ISO 6093, *Information processing - Representation of numeric values in character strings for information interchange.*

ISO 7942, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description.*

ISO 8632, *Information processing systems - Computer graphics - Metafile for the storage and transfer of picture description information*

- *Part 1 : Functional description*
- *Part 2 : Character encoding*
- *Part 3 : Binary encoding*
- *Part 4 : Clear text encoding*

ISO 8805, *Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) functional description.*

ISO/IEC 9593, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings.*

CIE Recommendations on colour space, supplement to CIE publication 15.

CIE 1976 Supplementary standard colour metric of server and coordinate systems.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

3 Definitions

For the purpose of this part of ISO/IEC 9592 the following definitions apply.

NOTE - As far as possible, graphics terminology which is commonly accepted and consistent with other graphics Standards is used.

3.1 acknowledgement: Output to the *operator* of a *logical input* device indicating that a *trigger* has fired.

3.2 addressable point: Any point of a device that can be addressed.

3.3 ancestor structure: A *parent structure* or the ancestor of a *parent structure*.

3.4 annotation: A class of *output primitives* that are defined in *normalized projection coordinates* but are placed with respect to a reference point which may be anywhere in *modelling coordinate space*. The plane on which the annotation appears is always parallel to the x-y plane of the *display space* and is unaffected by modelling and viewing transformations, but the reference point is transformed in the normal manner.

3.5 annotation style: An aspect of *annotation* indicating how relationships between an annotation primitive and a reference point are displayed.

3.6 annotation text relative: An *output primitive* consisting of a character string which is always drawn parallel to the x-y plane of the *display space*. Its position is determined by a reference point defined in *modelling coordinate space* and an offset in *normalized projection coordinates*.

3.7 application data: Data used by an application program, the nature of which is not specified in this standard. Application data is inserted into a *structure* as an "application data" *structure element*.

3.8 archive file: A mechanism for the storage and transportation of graphical data, represented by *PHIGS structures* and their contents.

3.9 aspects of output primitives: The appearance of *output primitives* is controlled by the values of a set of characteristics called "aspects" examples of which are the height of a character or the *linetype* of a *polyline*. Geometric aspects are *workstation* independent and are controlled by the corresponding *attributes*. For non-geometric aspects, the mapping between a particular aspect and its controlling *attribute* is defined by an associated *aspect source flag* (ASF). If the ASF is set to BUNDLED this aspect of the *output primitive* is controlled by the *bundle index attribute*. If the ASF is set to INDIVIDUAL then the aspect is controlled by the corresponding attribute.

3.10 aspect ratio: The ratio of lengths along the principal axes of an object.

3.11 aspect source flag (ASF): A flag indicating whether a particular *workstation* dependent *aspect of an output primitive* is selected from an *attribute* bundle, or as an individual *attribute* selection.

3.12 attribute: Attributes control the properties of *output primitives*. There are four types of attributes: geometric, non-geometric, viewing and identification. The geometric and non-geometric attributes control the values of *aspects of output primitives*.

3.13 back plane: A plane parallel to the *view plane* whose location is specified as an N coordinate value in the *view reference coordinate system*. *Output primitives* behind the back plane lie outside the *view volume*.

3.14 break action: An implementation dependent and *workstation* dependent mechanism enabling the *operator* to interrupt an input operation.

3.15 bundle index: An *attribute* of an *output primitive* which is an index into a *bundle table*; which defines the *workstation* dependent *aspects of the output primitive*.

3.16 bundle table: A *workstation* dependent table specifying aspects of one or more *output primitives*. *PHIGS* has *polyline*, *polymarker*, *text*, *interior*, and *edge* bundle tables.

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

3.17 bundle table entry: A single entry in a *bundle table*. Each entry contains one value for each aspect which applies to the corresponding *output primitive*. This set is *workstation* dependent.

3.18 cell array: An *output primitive* consisting of a parallelogram of equal sized cells, each of which is a parallelogram and has a single colour.

3.19 centralized structure store (CSS): The conceptual *workstation* independent storage area for *structure networks*.

3.20 character base vector: An aspect of *text* which defines the direction of the baseline of a character. It is a two-dimensional vector in the *text plane* specified in the *''text'' structure element*.

3.21 character body: The rectangle defining the horizontal and vertical limits of an individual character.

3.22 character expansion factor: An aspect of *text* which specifies the deviation of *character width* from the defined *nominal value* of a given font on a given *workstation*.

3.23 character height: An aspect of *text* which specifies the *nominal value* for the height of an upper case character.

3.24 character set: A registered interpretation for entries in the character code table (see ISO 2022).

3.25 character spacing: An aspect of *text* which specifies the fraction of the font *nominal value* for *character height* to be added between adjacent *character bodies* in a *string*.

3.26 character up vector: An aspect of *text* which defines the principal up direction of the text string. It is a two-dimensional vector in the *text plane* specified in the *"text" structure element*.

3.27 character width: An aspect of *text* which specifies the *nominal value* of the width of a character. The actual width depends on the width to height ratio specified by the designer of the font to which the character belongs.

3.28 child structure: A *structure* specified in a *structure reference*.

3.29 choice device: A *logical input device* providing a non-negative integer defining one of a set of alternatives.

3.30 CIE: Abbreviation for the Commission Internationale de l'Eclairage. Used to refer to the CIE universal colour definition system used as a colour model.

3.31 CIELUV: Abbreviation for the *CIE* 1976 (L*u*v*) colour space.

3.32 clipping: Removing parts of *output primitives* which lie outside a specified volume. The exact effect of clipping some *output primitives* may be implementation or *workstation* dependent.

3.33 colour index: An index used to access an entry in a *colour table*.

3.34 colour model: Characterization of a *colour space* in terms of explicit parameters.

3.35 colour space: Geometric representation of colours in space, usually of three dimensions.

3.36 colour system: A colour coordinate system.

3.37 colour table: A *workstation* dependent table, in which the entries specify the values defining a particular colour.

3.38 composite modelling transformation: A transformation applied to *output primitives* produced during structure *traversal*. It is defined as the concatenation of the *local modelling transformation* and *global modelling transformations* such that the *local modelling transformation* is the first transformation to be applied.

3.39 conflict resolution flag: During the process of *structure archiving* from the *CSS* or *structure retrieval* from an archive, naming conflicts may occur between *structures* on the *archive file* and *structures* in the CSS. The conflict resolution flag indicates how these conflicts will be resolved.

3.40 connection identifier: An implementation specific means of defining the connection to one or more physical entities which constitutes a single *workstation*.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Definitions

3.41 data record: A compound data type, the content of which is defined by the context within which it is used. For example, the content of the data records used in the input device initialization functions may vary depending upon the particular *prompt* and *echo type* specified in the invocation of the initialization function.

3.42 deferral mode: The deferral mode for a *workstation* is part of the *display update state* and specifies when changes to posted *structure networks* and the *workstation state list* shall be reflected in the displayed *image*.

3.43 descendant structure: A *child structure* or the descendant of a *child structure*.

3.44 device coordinates (DC): A device dependent coordinate system. In *PHIGS*, DC units are metres on a device capable of producing a precisely scaled *image*, and appropriate *workstation* dependent units otherwise.

3.45 device driver: The device dependent part of a *PHIGS* implementation that supports a physical graphics device. The device driver generates device dependent output and handles device dependent interaction.

3.46 device space: The space defined by the *addressable points* of a *display device*. *(Taken from ISO 7942 and ISO 8805)*

3.47 display device: A graphics device on which *images* can be represented. A display device is one possible component of a *workstation*.

3.48 display priority: The priority assigned to a *structure network* when it is posted. It is used to discriminate between *output primitives* when they are mapped to the same *display space* location.

3.49 display space:

(1) That portion of *device space* corresponding to the volume available for displaying *images*.

(2) The working space of an input device. *(Taken from ISO 7942 and ISO 8805)*

3.50 display surface: The physical area on a *display device* onto which *PHIGS images* may be placed.

3.51 display update state: Determines how and when the *display surface* is modified to reflect changes in the *centralized structure store* and the *workstation state list*. An application selects the display update state to take into account the capabilities of a *workstation* and the requirements of the application program. The display update state consists of the two *workstation* dependent aspects of *deferral mode* and *modification mode*.

3.52 echo: The immediate notification to the *operator* of the current *measure* of a *logical input device*.

3.53 echo area; echo volume: An area or volume, defined in *device coordinates*, which may be used for the display of a *prompt* or *echo*.

3.54 echo type: A parameter of device initialization which selects the *echo* technique for a particular *logical input device*.

3.55 edge: The set of boundaries of the polygons defined in the *fill area set output primitive*.

3.56 edge flag: An aspect of *fill area set* which enables or disables the display of *edges*.

3.57 edgetype: An aspect of *fill area set* which indicates the style of the *edges*.

3.58 edgewidth scale factor: An aspect of *fill area set* which indicates the relative width of the *image* of an *edge*. The edgewidth scale factor is applied to a *workstation* dependent *nominal value*.

3.59 edit mode: Determines whether a new *structure element* will replace the *structure element* at the *element pointer* or will be inserted into the open *structure* after the *element pointer*.

3.60 element pointer: A pointer used during *structure editing*, the value of which identifies the position in the open *structure* at which element deletion and creation will occur.

3.61 element position: A number associated with a *structure element* which indicates the element's position within a *structure*.

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

3.62 element reference list: A list of references which define the hierarchy within one branch of a *structure network*. Each reference consists of a *structure identifier* and an *element position* within that *structure*. If the list contains N pairs, then the first N-1 pairs identify EXECUTE STRUCTURE elements. The jth such element references the structure named in the (j+1)st list element. The deepest element of the list may identify any type of structure element.

3.63 element type: The identifying classification of a *structure element*. For example, *fill area*, *label*, *application data*, *linewidth scale factor*.

3.64 empty interior style: One possible representation of the *interior* of a *fill area* or *fill area set output primitive*. If the *edges* are not displayed, the *image* of a *fill area set* with *interior style* empty is invisible. The *image* of a *fill area output primitive* with *interior style* empty is always invisible.

3.65 error state list: The data holding information about the most recent error condition.

3.66 escape: A function which provides access to implementation dependent or device dependent features not concerned with the generation of graphical output.

3.67 event mode: An operating mode for a *logical input device* in which asynchronous input is placed on the *event queue* as an *event report* when a *trigger* fires.

3.68 event queue: A time-ordered collection of *event reports*.

3.69 event report: An entry in the *event queue* which consists of a logical input value and identification of the logical input device responsible.

3.70 exclusion set: The portion of a *filter* which defines those *name set* members which are not eligible for a certain operation.

3.71 fill area: An *output primitive* consisting of a single polygon.

3.72 fill area set: An *output primitive* consisting of a set of *fill areas* with or without *edges*.

3.73 filter: The combination of the *inclusion set* and the *exclusion set* which identify *output primitives* eligible or ineligible for a certain operation. *PHIGS* supports filters for *picking*, *highlighting*, *invisibility*, and incremental spatial search.

3.74 font: A set of character representations all of which share certain visual characteristics.

3.75 front plane: A plane parallel to the *view plane* which is specified as an N coordinate value in the *view reference coordinate system*. *Output primitives* in front of the front plane lie outside the *view volume*. *(Taken from ISO 8805)*

3.76 generalized drawing primitive (GDP): An *output primitive* which accesses implementation dependent and *workstation* dependent geometric capabilities such as curve drawing.

3.77 generalized structure element (GSE): A *structure element* which is used to access implementation dependent, *workstation* dependent, or device dependent features during structure *traversal*. It is a *structure element* which accesses *attribute* or control functionality, but does not create an *output primitive*.

3.78 GKS: Graphical Kernel System (ISO 7942)

3.79 global modelling transformation: A component of the *composite modelling transformation*. When *traversal* of a *structure* begins, it is set to the current *composite modelling transformation* of the *parent structure* or if it is the posted *structure*, the default value in the *PHIGS description table*.

3.80 hatch interior style: One possible representation of the *interior* of a *fill area set* or *fill area output primitive*. The *interior* is filled with a pattern of parallel and/or crossing hatch lines, selected from the *workstation*'s *hatch table*.

3.81 hatch table: The table of the hatch values defined on a *workstation*.

3.82 hidden line/hidden surface removal (HLHSR): Removal of those parts of *output primitives* which are obscured by other *output primitives*.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Definitions

3.83 highlighting: Emphasizing an *output primitive* by modifying its visual *attributes* in some *workstation* dependent manner.

3.84 highlighting filter: A *filter* consisting of two *name sets*, the highlighting *inclusion set* and the highlighting *exclusion set*, used to identify *output primitives* which are eligible for *highlighting*.

3.85 HLS: An abbreviation for the Hue, Lightness, Saturation *colour model*.

3.86 hollow interior style: One possible representation of the *interior* of a *fill area* or *fill area set output primitive*. The *image* is the boundary line only, including any boundaries created by *clipping*.

3.87 HSV: An abbreviation for the Hue, Saturation, Value *colour model*.

3.88 image: The appearance of objects after rendering.

3.89 implicit regeneration: The complete recreation of the contents of a *display surface* such that it is *visually correct*. This may occur when changes to the posted *structure networks* or the *workstation state list* invalidate the displayed *image*. Such a regeneration is not explicitly requested by the application program.

3.90 inclusion set: The portion of a *filter* which defines those *name set* members eligible for a certain operation.

3.91 inheritance: The mechanism by which *child structures* obtain initial *attribute* settings from their *ancestor structures*.

3.92 input class: The characterization of the functionality of a logical input device. There are six classes of logical input device: locator, stroke, valuator, choice, pick, and string.

3.93 input mode: One of the three possible methods of obtaining data from a logical input device: REQUEST, SAMPLE, or EVENT.

3.94 inquiry function: A mechanism for communicating to the application program data contained in a state list or description table.

3.95 interior: The set of points which lie inside a *fill area* or *fill area set*. *PHIGS* provides a rule for determining whether or not a given point is inside a *fill area* or *fill area set*.

3.96 interior style: An aspect which indicates the style used to fill the *interior* of a *fill area* or *fill area set*.

3.97 invisibility: The state of an *output primitive* being hidden from view even when lying within the *display surface* and not occluded by other *output primitives*.

3.98 invisibility filter: A *filter* consisting of two *name sets*, the *invisibility inclusion set* and the *invisibility exclusion set*, used to identify *output primitives* eligible for *invisibility*.

3.99 isotropic mapping: A transformation which preserves *aspect ratio*.

3.100 label: A *structure element*, consisting of an identifier, which can be used as a place-marker to facilitate *structure editing*.

3.101 language binding: The expression of a functional specification in the syntax of a particular programming language.

3.102 linetype: An aspect which indicates the style of the *image* of a *polyline*, such as solid, dashed or dotted.

3.103 linewidth scale factor: An aspect which indicates the relative width of the *image* of a *polyline*. The linewidth scale factor is applied to a *workstation* dependent *nominal value*.

3.104 local modelling transformation: A component of the *composite modelling transformation*. When *traversal* of a *structure* begins, it is set to the identity transformation.

3.105 locator device: A *logical input device* providing a position in *world coordinates* and an associated *view index*.

3.106 logical input device: An abstraction of one or more *physical input devices* which delivers logical input values to the application program.

Documents
Document Sharing Hub
Full document details: ISO_IEC 9592-1_1989

3.107 marker: A glyph with a specified appearance which is used to identify a location on a *display surface*. The shape of a marker is not subject to transformation.

3.108 marker size scale factor: An aspect which indicates the relative size of the *image* of a *marker*. The marker size scale factor is applied to a *workstation* dependent *nominal value*.

3.109 marker type: An aspect which selects the type of glyph used for *markers*.

3.110 measure: A value which is determined by the values of one or more *physical input devices* and a mapping from these values to a form required by one of the logical *input classes*. *(Taken from ISO 7942 and ISO 8805)*

3.111 measure process: A process which comes into existence when a *logical input device* is enabled for interaction. The current state of a measure process is the *measure*.

3.112 message: A *string* sent by the application program to a *workstation* principally for communication with the *operator*.

3.113 metafile: A mechanism for retaining and transporting graphical data and control information. A metafile contains a device independent description of one or more pictures. *(Taken from ISO 8632)*

3.114 minimal simulation: The minimal required action which is necessary when a *workstation* does not have the capability to render an *output primitive* as specified by the application.

3.115 modal: A type of change mechanism which extends its effect until another change specification supersedes it. *Attribute* settings and control functions are modal for a given *structure* and its *descendant structures*.

3.116 modelling coordinates: The device independent three-dimensional Cartesian coordinate system in in which graphical objects are defined by the application program using the *PHIGS output primitives*.

3.117 modelling transformation: See *local modelling transformation*, *global modelling transformation*, *composite modelling transformation*.

3.118 modification mode: Part of the *display update state*. The modification mode governs what category of visual effects shall be immediately achieved, and how they are achieved, on a *workstation*.

3.119 name set: An *output primitive attribute* consisting of a set of classifications defining the eligibility of *output primitives* for *highlighting*, *invisibility*, *picking* and incremental spatial search.

3.120 nominal value: The *workstation* dependent base value for *aspects of output primitives*.

3.121 normalized projection coordinates (NPC): The device independent three-dimensional Cartesian coordinate system in which the composition of images is specified to the graphics system. The view clipping limits and the *workstation window* are specified in NPC space.

3.122 operator: Person manipulating *physical input devices* so as to change the measures of *logical input devices* and cause their *triggers* to fire. *(Taken from ISO 7942 and ISO 8805)*

3.123 output primitive: A fundamental graphical display entity representing a geometric shape.

3.124 parallel transformation: The transformation of an object in which parallel lines in the object appear parallel in the resulting *image* without regard to relative distance or depth. *(Taken from ISO 8805)*

3.125 parent structure: A *structure* which contains one or more *structure references*.

3.126 pattern interior style: One possible representation of the *interior* of a *fill area* or *fill area set output primitive*. The *interior* is filled in with a two dimensional pixel pattern selected from the *workstation*'s *pattern table*.

3.127 pattern table: The table of the pattern values defined on a *workstation*.

3.128 perspective transformation: The transformation of an object in which parallel lines in the object which intersect the *view plane* appear to converge in the resulting *image* as a function of relative distance or depth. *(Taken from ISO 8805)*

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Definitions

3.129 PHIGS: The Programmer's Hierarchical Interactive Graphics System.

3.130 PHIGS description table: The data which represents the static *workstation* independent parameters of a *PHIGS* system.

3.131 PHIGS state list: The data which represents the dynamic *workstation* independent parameters of a *PHIGS* system.

3.132 PHIGS traversal state list: The data which represents the dynamic *workstation* independent parameters of a *PHIGS* system during *traversal* of a *structure network*. When *traversal* begins the *PHIGS* traversal state list values are set to the values defined in the *PHIGS description table*.

3.133 physical input device: Hardware capable of generating input. A physical input device is either part of a *display device* or a separate part of the *workstation*.

3.134 picking: An *operator* action of identifying an *output primitive* using a *pick device*.

3.135 pick device: A *logical input device* providing a *pick path*.

3.136 pick filter: A *filter* consisting of two *name sets*, the pick *inclusion set* and the pick *exclusion set* used to identify *output primitives* which are eligible for selection by a pick input operation.

3.137 pick identifier: A name associated with individual *output primitives*, and returned by a *pick device* as a component of the *pick path*.

3.138 pick path: The *element reference list* representing a path to a picked *output primitive* from its posted *structure*, consisting of the *structure identifier*, *pick identifier*, and *element position* for each level of hierarchy traversed.

3.139 polyline: An *output primitive* consisting of a connected sequence of straight lines between specified points.

3.140 polymarker: An *output primitive* consisting of a set of locations, each indicated by the same type of *marker*.

3.141 post: To identify a *structure network* for display on a particular *workstation*.

3.142 projection reference point (PRP): A point in *view reference coordinate space* which determines the direction of *projectors* when the *projection type* is PARALLEL, or from which all *projectors* emanate when the *projection type* is PERSPECTIVE. *(Taken from ISO 8805)*

3.143 projection type: The type of transformation to be used in view mapping, i.e., *parallel transformation* or *perspective transformation*.

3.144 projection viewport: A rectangular parallelepiped in *normalized projection coordinate* space. In the *PHIGS* viewing model, the view mapping transformation maps the contents of the *view volume* into the projection viewport.

3.145 projector: A conceptual line which passes through a point of an object and intersects the *view plane*.

3.146 prompt: Output to the *operator* indicating that a specific *logical input device* is available. *(Taken from ISO 7942 and ISO 8805)*

3.147 request mode: An operating mode for a *logical input device* in which *PHIGS* waits until the *operator* activates the appropriate *trigger* where upon the *measure* is returned.

3.148 RGB: An abbreviation for the Red, Green, Blue *colour model*.

3.149 sample mode: An operating mode for a *logical input device* in which the *measure* of a *logical input device* may be obtained immediately without waiting for a *trigger*.

3.150 solid interior style: One possible representation of the *interior* of a *fill area* or *fill area set output primitive*. The *interior* is filled with a uniform colour.

3.151 string: An ordered collection of character codes as described in ISO 2022.

AVA Documents 11
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

3.152 string device: A *logical input device* providing a *string*.

3.153 stroke device: A *logical input device* providing a sequence of points in *world coordinates*, and an associated *view index*.

3.154 structure: A linear sequence of *structure elements*.

3.155 structure archiving: The process of storing *structures* and *structure networks* in an *archive file* for subsequent *structure retrieval*.

3.156 structure editing: The modification of a *structure*, including its initial creation.

3.157 structure element: The fundamental unit of data in the *centralized structure store*, a sequence of which is collected to form a *structure*. Structure elements include elements causing the generation of *output primitives*, *attribute* selections, *labels*, *application data*, *NAME SET* specifications, transformation selections and *structure references*.

3.158 structure identifier: A unique application specified name used to reference a structure.

3.159 structure network: A collection of *structures* arranged with the topology of an acyclic directed graph, where *structures* are thought of as nodes and execute *structure elements* are thought of as connections from the *parent structure* to the *child structure*.

3.160 structure reference: The invocation of one *structure* from within another by an execute *structure element* or the invocation of a *structure* by posting.

3.161 structure retrieval: The process of copying *structures* from an *archive file* to the *centralized structure store*.

3.162 structure state list: The data which represents information about each *structure* in the *centralized structure store*.

3.163 text: An *output primitive* consisting of a character string drawn at a specified position on a specified *text plane*.

3.164 text alignment: An aspect of *text* which specifies the mode of justification. This aspect has horizontal and vertical components.

3.165 text direction vectors: Two vectors, specified as part of the "*text*" *structure element*, which, together with the *text position*, define the *text plane*.

3.166 text extent: A parallelogram defining the minimum area which completely encloses the *character bodies* in a *string*.

3.167 text font: An aspect of *text* which indicates certain visual properties of a character, such as typeface.

3.168 text local coordinate system: The local coordinate system for *text output primitives* within which all geometric *text attributes* are measured.

3.169 text path: An aspect of *text* which controls the relative positioning of successive characters in a *string*.

3.170 text plane: The plane in which a *text output primitive* appears.

3.171 text position: A point, specified as part of the "*text*" *structure element*, which, together with the *text direction vectors*, determine the *text plane*. It also determines, together with *text alignment*, the location of the *string*.

3.172 text precision: An aspect of *text* which selects the fidelity with which the drawing of a *text output primitive* matches the appearance selected by an application program.

3.173 transformation pipeline: The ordered sequence of transformations which convert *modelling coordinates* to *device coordinates*.

3.174 traversal: The process of stepping through a *structure network*, elaborating each *structure element* by generating an *output primitive* or changing the state lists.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Definitions

3.175 traversal state list: See *PHIGS traversal state list*.

3.176 trigger: A *physical input device* or set of devices that an *operator* can use to indicate significant moments in time. *(Taken from ISO 7942 and ISO 8805)*

3.177 trigger process: A process which notifies an associated *logical input device* when a *trigger* is fired. The trigger process only exists when a device is in *EVENT mode*, or in *REQUEST mode* with a request pending.

3.178 unpost: To remove the identification of a *structure* for display on a particular *workstation*.

3.179 valuator device: A *logical input device* providing a real number.

3.180 view index: An *output primitive attribute* that selects the *view representation* applied to *output primitives* and geometric *attributes* during structure *traversal*.

3.181 view mapping matrix: A 4×4 matrix which specifies the transformation of points from *view reference coordinates* to *normalized projection coordinates*.

3.182 view orientation matrix: A 4×4 matrix which specifies the conversion of points from *world coordinates* to *view reference coordinates*.

3.183 view plane: A plane parallel to the U-V plane in *view reference coordinate space (VRC)*. It is specified as an N coordinate value in VRC.

3.184 view plane normal: A vector in *world coordinates*, relative to the *view reference point*, which defines the N coordinate axis of the *view reference coordinate system*.

3.185 view reference coordinate (VRC) system: A device independent three-dimensional Cartesian coordinate system in which the parameters for the view mapping transformation are specified. The axes of the view reference coordinate system are called the U, V and N axes. The position and orientation of this coordinate system, relative to *world coordinates*, are defined by the *view reference point*, the *view plane normal* and the *view up vector*. *(Taken from ISO 8805)*

3.186 view reference point: A point in *world coordinates* which defines the *view reference coordinate system* origin. *(Taken from ISO 8805)*

3.187 view representation: A single entry in the *view table*. It consists of the *view orientation matrix*, the *view mapping matrix*, the view clipping limits and a set of clipping indicators.

3.188 view table: The table of *view representations* defined on a *workstation*. Its entries are referenced by the *view index attribute*.

3.189 view transformation input priority: Determines the order in which *view table* entries are tested when selecting the inverse viewing transformation to be applied to locator and stroke input.

3.190 view up vector: A vector defined in *world coordinate* space. The *view reference coordinate system* V coordinate axis is defined as an orthogonal projection of the view up vector onto the plane through the *view reference point* and perpendicular to the *view plane normal*. Vectors which are parallel to the view up vector in *world coordinate* space will appear vertical in the final *image*. *(Taken from ISO 8805)*

3.191 view volume: The view volume is determined by the *view window* defined in the *view plane*, the *front plane*, the *back plane*, and *projectors* through the corners of the *view window*. If the *projection type* is PERSPECTIVE, the *projectors* all pass through the *projection reference point*, the view volume is therefore a frustum. If the *projection type* is PARALLEL, the projectors are parallel to the vector joining the *projection reference point* to the centre of the *view window*, the view volume in this case is a parallelepiped. In two dimensional application programs, the view volume reduces to the *view window*. *(Taken from ISO 8805)*

3.192 view window: A rectangle in the *view plane*. *Projectors* passing through the corners of the view window define the left, right, bottom and top surfaces of the *view volume*. *(Taken from ISO 8805)*

3.193 visually correct: The display on a *workstation* is visually correct when it corresponds exactly to the state of *centralized structure store* and the *workstation state list*.

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

3.194 workstation: *PHIGS* is based on the concept of abstract graphical resources, which provide the logical interface through which the application program controls physical devices. A workstation consists of zero or one *display space* and zero or more *logical input devices*.

3.195 workstation category: A property of a *workstation* which indicates whether it can perform graphics input only, graphics output only, or both.

3.196 workstation description table: The data which represents capabilities for a particular *workstation type*.

3.197 workstation identifier: A unique application specified name used to identify a particular *workstation*.

3.198 workstation state list: The data which represents dynamic information for a particular *workstation*.

3.199 workstation transformation: A transformation which maps the contents of the *workstation window* into the *workstation viewport*, preserving *aspect ratio* in X and Y, but not necessarily in Z.

3.200 workstation type: A type or class of actual *workstation*, sharing common characteristics and a single *workstation description table*.

3.201 workstation viewport: A rectangular parallelepiped in *device coordinate* space. All graphics output appears within this volume. *(Taken from ISO 7942 and ISO 8805)*

3.202 workstation window: A rectangular parallelepiped in *normalized projection coordinate* space. Its contents are mapped by the *workstation transformation* into the *workstation viewport*. *(Taken from ISO 8805)*

3.203 world coordinates (WC): The device independent three-dimensional Cartesian coordinate system used by the application program to organize modelled objects for display. The effect of applying the *composite modelling transformation* to *modelling coordinates* is to produce world coordinates.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

4 The Programmer's Hierarchical Interactive Graphics System

4.1 About this part of ISO/IEC 9592

4.1.1 Specification and conformance

The set of functions known as PHIGS shall be as described in clauses 4, 5 and 6. A conforming implementation of PHIGS shall correctly implement all the functions described. Minimum support criteria are specified in table 1 (4.14). A conforming PHIGS implementation shall also provide the information described in annex D and the minimum support specified in 4.3. In a conforming implementation all graphical capabilities that can be addressed by PHIGS shall be used only via PHIGS.

4.1.2 Registration

For certain parameters of the functions, PHIGS defines value ranges as being reserved for registration. 1) The meanings of these values will be defined in the register. These procedures do not apply to values and value ranges defined as being workstation or implementation dependent; these values and ranges are not standardized.

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 JTC 1 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.

Documents
Document Sharing Hub
Full document details: ISO_IEC 9592-1_1989

4.2 Overview

The Programmer's Hierarchical Interactive Graphics System (PHIGS) provides a functional interface between an application program and a configuration of graphical input and output devices. The functional interface contains basic functions for dynamic interactive hierarchical graphics on a wide variety of graphics equipment.

The interface is at such a level of abstraction that hardware peculiarities are shielded from the application program. As a result, a simplified interface presenting uniform output primitives (POLYLINE, POLYMARKER, TEXT, ANNOTATION TEXT RELATIVE, FILL AREA, FILL AREA SET, CELL ARRAY, GENERALIZED DRAWING PRIMITIVE), and uniform input classes (LOCATOR, STROKE, VALUATOR, CHOICE, PICK, STRING) is obtained.

In 4.4 the concepts of structures and the centralized structure store are introduced.

In 4.5 and 4.8 the concepts of basic output, input and the organization of input and output sequences are outlined.

In 4.6 a central concept, the workstation, is introduced both for structuring PHIGS and for realizing device independence. The concept of multiple workstations allows simultaneous output to and input from various display systems.

Some PHIGS concepts are described as being *workstation dependent*. This means that an implementation is allowed to specify the characteristics of these concepts on a workstation by workstation basis.

All workstation dependent concepts are explicit in the standard. Other PHIGS concepts are described as being *implementation dependent*. This means that an implementation is allowed to determine the effect (rather than having the effect mandated by the standard). However, such effects shall be the same on all workstations in a single implementation.

Annex D describes allowable differences between PHIGS implementations.

PHIGS defines only a language independent nucleus of a graphics system. For integration into a language, PHIGS is embedded in a language dependent layer containing the language conventions, for example, parameter and name assignment.

The layer model represented in figure 1 illustrates the role of PHIGS in a graphics system. Each layer may call the functions of the adjoining lower layers. In general the application program uses the application oriented layer, the language dependent layer, other application dependent layers, and operating system resources.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

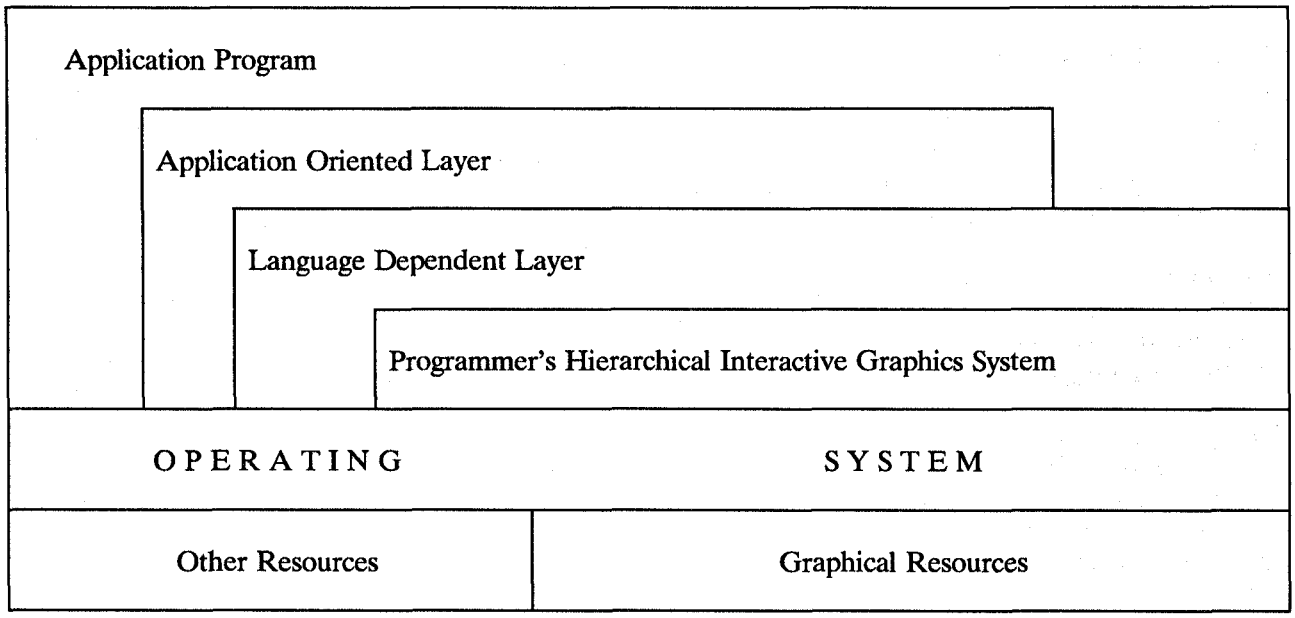


Figure 1 - Layer model of PHIGS

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

4.3 Concepts

4.3.1 PHIGS concepts

PHIGS supports the storage and manipulation of data in a centralized hierarchical data structure, known as the *centralized structure store* (CSS). The fundamental entity of data is a *structure element* and these are grouped together into units called *structures*. Structures are organized as acyclic directed graphs called *structure networks*. The creation and manipulation of the data structure is independent of its display.

Structure elements can be either graphical data or application specific data. Graphical data consists of information for structure management and control as well as for output definition. Functions are provided to edit structures by the insertion and removal of structure elements. Functions are also provided to support searching and inquiry of the content and topology of structure networks and the CSS in general.

The graphical output that is generated by PHIGS is built up from two groups of basic elements called *output primitives* and *primitive attributes*. The output primitives are abstractions of basic actions a device can perform, such as drawing lines and displaying character strings. The attributes control the properties of the primitives on a device, such as their aspects; for example linetype, colour, and character height. Non-geometric aspects such as colour can be controlled for each workstation separately to make best use of its capabilities.

The graphical information that is input from a device as a result of operator actions is mapped by PHIGS onto six classes of input, each represented by a data type referred to as a logical input value. An instance of such a device representation is called a logical input device. The effect of input actions on the display surface, such as prompts and echoes, is controlled by PHIGS for each logical input device individually.

The two abstract concepts (abstract output and abstract input) are the building blocks of an abstract workstation. A PHIGS workstation represents a unit consisting of zero or one display surfaces and zero or more input devices, such as keyboard, tablet and lightpen. The workstation presents these devices to the application program as a configuration of abstract devices thereby shielding the hardware peculiarities.

Graphical output on a workstation is produced by traversing a structure identified for display on that workstation and interpreting the structure elements. Application specific data is generally ignored during structure traversal.

The geometric information (coordinates) stored within the graphical structure elements are processed through workstation independent and workstation dependent stages. The workstation independent stage of structure traversal performs a mapping from modelling coordinates to a world coordinate system during which graphical structure elements can produce output primitives.

The workstation dependent stage then performs a transformation on the geometric information contained in output primitives, attributes and logical input values (locator and stroke). These transformations perform mappings between four coordinate systems, namely:

a) World Coordinates (WC) used to defined a uniform coordinate system for all abstract workstations;

b) View Reference Coordinates (VRC), used to define a view;

c) Normalized Projection Coordinates (NPC) used to facilitate assemblies of different views;

d) Device Coordinates (DC), one coordinate system per workstation representing its display space.

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Output primitives and attributes are mapped from WC to VRC by the view orientation transformation, from VRC to NPC by the view mapping transformation and finally from NPC to DC by the workstation transformation. Locator and stroke input is mapped by the inverse of these transformations from DC to WC.

Regions may be identified in the world and normalized projection coordinate systems which, during the mappings described above, may result in the clipping of portions of graphical output which lie outside one or more of the regions. Modelling clip regions may be defined in modelling coordinates. These regions are then transformed to world coordinates and combined to form a composite modelling clip region which may then be applied to the graphical output. Another clip region is defined as part of a view definition to limit the generation of graphical output. A third clip region associated with the workstation transformation identifies a subset of the image for display on the workstation.

The attributes which control the appearance of parts of the picture (output primitives, prompt and echo types) on the display space are organized in a uniform manner. Two groups of attributes apply to the appearance of each output primitive: primitive attributes (that are workstation independent) and workstation attributes. Primitive attributes are specified modally and are bound to a primitive when it is created at structure traversal time. The primitive attributes include all geometric aspects of primitives, such as character height for text and pattern size for fill area and fill area set. In addition, the non-geometric aspects of primitives are controlled by the primitive attributes in one of two ways. Either an attribute is used to specify a group of the non-geometric aspects of the primitive by an index (or set of indices in the case of the fill area primitives) which points to a workstation dependent representation (set of values) or one attribute is used to specify each of the non-geometric aspects of the primitive in a workstation independent way. The former is referred to as bundled specification and the latter is referred to as individual specification. In addition, when a primitive is created by traversal an index is bound to it which selects the view orientation and view mapping transformations which are workstation dependent.

Each primitive has associated with it a NAME SET attribute defined at structure traversal time. This NAME SET attribute defines the eligibility of the primitive for highlighting, invisibility, picking and incremental spatial search. A filter mechanism is incorporated in PHIGS with an inclusion and exclusion set mask at each workstation to control specific rendering of the primitives.

The attribute selection is controlled by the nature of the structure network. The effect of an attribute is defined within the structure in which it occurs and in subsequent structures below it in the hierarchy. Output renderings of the same structure may therefore be different due to the inheritance of attributes.

Workstation attributes include representations for those non-geometric aspects which are workstation dependent, and are selected via bundle indices. For example, the representations (or bundles) for polyline each contain values of linetype, linewidth scale factor and colour index. Workstation attributes also specify the colour and pattern tables and control the updating of the display. Workstation attributes can be reset dynamically.

Hidden lines or hidden surfaces may be removed using a workstation dependent method (see 4.5.14 and 4.6.6).

The attributes which control the operation of logical input devices can be specified either upon initialization or as part of input device setting, depending upon the attributes. Through initialization, an initial value, a prompt and echo technique, and an area on the device for echoing can be specified. A data record may further provide device specific attributes. Through input device setting, the operating mode may be selected and the echo may be switched on or off. The operating modes of logical input devices specify who (operator or application program) has the initiative:

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

e) SAMPLE input is acquired directly by the application program;

f) REQUEST input is produced by the operator in direct response to the application program;

g) EVENT input is generated asynchronously by the operator and is collected in a queue for the application program.

PHIGS provides an interface to enable the information contained in the centralized structure store to be archived or retrieved. These PHIGS *archive files* provide for the storage and interchange of graphics models expressed as PHIGS structure networks.

PHIGS also provides an interface to a system for filing graphical image information for the purpose of external long term storage and exchange of graphical image information. The interface consists of a metafile output workstation, which writes to a graphics metafile, and a metafile input workstation, which reads from the metafile. In addition to the normal functions for output to workstations, a metafile workstation may accept items containing non-graphical information. Input from a metafile is controlled by read and interpret functions.

4.3.2 Relationship to ISO 7942 (GKS) and ISO 8805 (GKS-3D)

The primary emphasis of PHIGS is different from that of GKS/GKS-3D. Unlike GKS/GKS-3D, PHIGS is not restricted to graphics and provides data structure definition and editing facilities which are not necessarily related to graphical output. PHIGS defines data structures which can be rendered as graphical output, rather than being purely directed towards providing graphical output and interaction.

Whereas GKS provides for graphical output and interaction in a 2D space, GKS-3D and PHIGS provide graphical output and interaction in a 3D space. The relationship between GKS and GKS-3D is necessarily closer than that between GKS and PHIGS, and whereas GKS-3D is oriented towards the direct output of graphical primitives grouped into segments, PHIGS provides data structures which are traversed to produce graphical output. However, the output primitives and their attributes generated at structure traversal time are a superset of those of GKS-3D.

PHIGS provides the same underlying model of input as GKS-3D. The measure of the pick logical input device is extended to identify the structure element in the CSS which generates the primitive picked by the operator.

The original GKS concept of the workstation has been extended to support the increased dimensionality. Also, PHIGS supports posting to workstations rather than workstation activation for the purpose of directing graphics output to a workstation.

Annex F describes the technical differences between PHIGS and GKS/GKS-3D.

4.3.3 Notational conventions

This Standard uses the following typographical conventions.

a) The names of attributes of primitives appear in upper case (see 4.5.2).

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

b) Geometric aspects of primitives appear in upper case. Non-geometric aspects appear in lower case except when the corresponding individually specified attributes are being used, in which case they appear in upper case (see 4.5.2).

c) The names of entries in the PHIGS data structures (e.g. state lists and description tables) appear in lower case surrounded by single quotation marks.

d) The names of structure elements appear in lower case surrounded by double quotation marks.

e) The names of PHIGS functions appear in upper case.

f) The values of an enumeration data type appear in upper case (see 6.1).

g) The names of data types appear in upper case.

Documents21
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

4.4 The centralized structure store

4.4.1 Structure elements and structures

PHIGS supports the storage and manipulation of data in a *centralized structure store* (CSS). The CSS contains graphical and application data organized into units called *structures*, which may be related to each other hierarchically to form *structure networks*. Each structure is identified by a unique name which is specified by the application.

A structure consists of a sequence of *structure elements*, where a structure element is the fundamental entity of data. Structure elements are used to represent application specified graphics data for output primitives, attribute selections, view selections, modelling transformations and clipping, name sets, invocations of other structures, and to represent application data. In addition, generalized structure elements can be used to access additional, implementation dependent features. Label structure elements may be used to identify locations in structures to assist in editing. Structure elements cannot exist outside of structures.

The following structure elements are defined in PHIGS.

Output primitive structure elements

polyline 3	fill area 3
polyline	fill area
polymarker 3	fill area set 3
polymarker	fill area set
text 3	cell array 3
text	cell array
annotation text relative 3	generalized drawing primitive 3
annotation text relative	generalized drawing primitive

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Attribute specification structure elements

set polyline index	set annotation text character up vector
set polymarker index	set annotation text path
set text index	set annotation text alignment
set interior index	set annotation style
set edge index	set interior style
set linetype	set interior style index
set linewidth scale factor	set interior colour index
set polyline colour index	set edge flag
set marker type	set edgetype
set marker size scale factor	set edgewidth scale factor
set polymarker colour index	set edge colour index
set text font	set pattern size
set text precision	set pattern reference point and vectors
set character expansion factor	set pattern reference point
set character spacing	add names to set
set text colour index	remove names from set
set character height	set individual ASF
set character up vector	set hlhsr identifier
set text path	set view index
set text alignment	set pick identifier
set annotation text character height	

Transformation and clipping structure elements

set local transformation 3	set modelling clipping volume 3
set local transformation	set modelling clipping volume
set global transformation 3	set modelling clipping indicator
set global transformation	restore modelling clipping volume

Control structure element

execute structure

Editing structure element

label

Generalized structure element

generalized structure element

Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

Application data element

application data

Although the PHIGS CSS is conceptually a centralized data model, this does not preclude an implementation from decentralizing the physical organization of the data storage, provided that this implies no restrictions on the functionality.

4.4.2 Structure networks

Structure networks are organized as directed acyclic graphs. That is, a structure may contain invocations of other structures contained within the CSS. The invocation of a structure is achieved using the "*execute structure* " element. Such an invocation is known as a *structure reference*. Structure networks cannot be recursive: there should not exist a sequence of invocations from a structure which results in the invocation of that structure; however, it is not mandated that this condition be checked.

A PHIGS structure network can be represented diagrammatically as shown in figure 2. Here, the structures are represented by circles labelled with the structure identifiers, and structure references are shown by arrows, which point to the invoked structures.

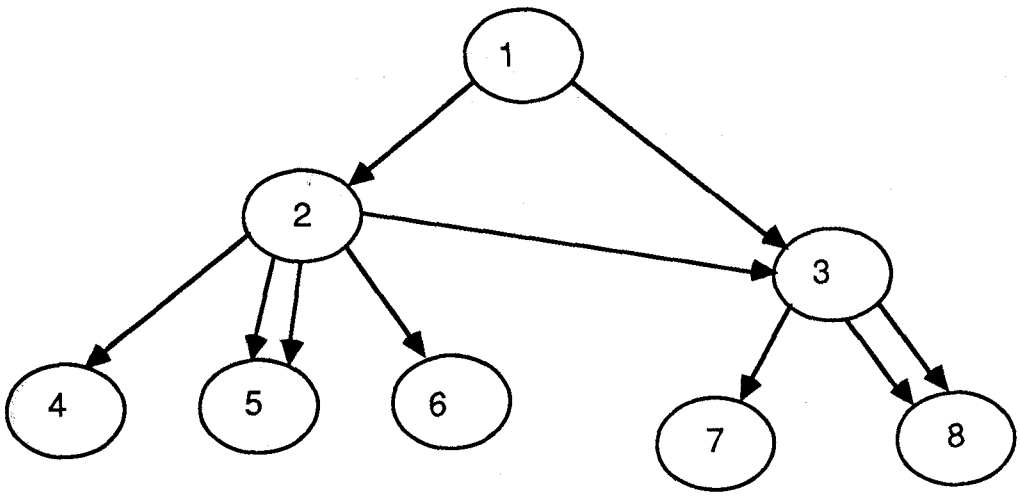


Figure 2 - A hierarchical structure network

When a structure S contains invocations of other structures, it is said to be a *parent* of those structures; correspondingly, each of the invoked structures is said to be a *child* of S. For example, structure 2 is the parent of structures 4,5, and 6; structures 2 and 3 are children of structure 1. Also, structures 1 and 2 are said to be *ancestors* of structures 4,5 and 6; while structures 7 and 8 are *descendants* of structures 1, 2 and 3.

Many types of operations available to the application are allowed on structure networks within the CSS. These are structure display, structure editing, structure manipulation, search and inquiry, and structure

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

archival and retrieval. Conceptually, all changes to the CSS, which result from any of the above operations are atomic and take effect immediately. However, the actual time at which visual changes take effect depends on the capabilities of the workstation and the current display update state (see 4.6.3).

For all of the structure operations, a structure exists if and only if the PHIGS CSS has knowledge of the structure's unique identifier. A structure can be created in one of the following ways:

a) when a reference to non-existent structure is inserted into a structure in the CSS;

b) when the structure is opened for the first time;

c) when the structure is posted for display on a workstation;

d) when the structure is referenced in any function changing the structure identifier;

e) when a structure not existing in the CSS is retrieved from an archive;

f) when a non-existent structure is emptied.

The following functions take no action if a specified structure is non-existent:

g) an unpost operation;

h) an operation to copy all elements from a non-existent structure into the currently open structure;

i) a delete structure operation.

4.4.3 Structure traversal and display

A structure network is identified for display on a workstation by the posting function, POST STRUCTURE. A structure may be displayed only if it is a member of a posted structure network (see figure 3).

When a non-existent structure is posted, a new empty structure is created and posted. The functions UNPOST STRUCTURE and UNPOST ALL STRUCTURES are used to disassociate structure networks from a workstation. The structure display functions affect the 'table of posted structures' in the workstation state lists. The actual display of the structures identified in the 'table of posted structures' is subject to display update state (see 4.6.3).

To display a network, the structure elements have to be extracted from the centralized structure store and processed. *Traversal* is an operation used to describe a method of structure element processing. Traversal of a network leads to the display of graphical output from the structure network at each workstation to which the network is posted. Conceptually, traversal is a process which is associated with each instance of a posted structure network. The process remains in existence until the structure network is unposted from the workstation through UNPOST or DELETE functions.

The traversal process interprets each structure element in the structure network sequentially, starting at the first element of the top of the network. Any change to the posted structure network triggers a retraversal of the structure network, so that any changes to a posted structure network are processed immediately. Whether the resulting changes are visible at the workstation depends on the display update state of the workstation (see 4.6.3). No knowledge of how traversal takes place is required. Its effect is as described above. How the traverser chooses to achieve the effect is implementation dependent.

A *display priority* is assigned to a structure network when it is posted. The application indicates relative importance of posted structure networks through this display priority. If multiple structures are posted for display to the same display space location the implementation will ensure the display of the higher priority structure network. Where two structures have the same priority, an implicit priority based on the

Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

sequence of posting is used, whereby the last posted structure network has the higher priority.

A PHIGS *traversal state list* is associated with each traversal process. Values in this state list may be accessed or modified when structure elements are interpreted. Each time a traversal is initiated, the associated PHIGS traversal state list is initialized. The initial values are taken from the PHIGS description table.

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

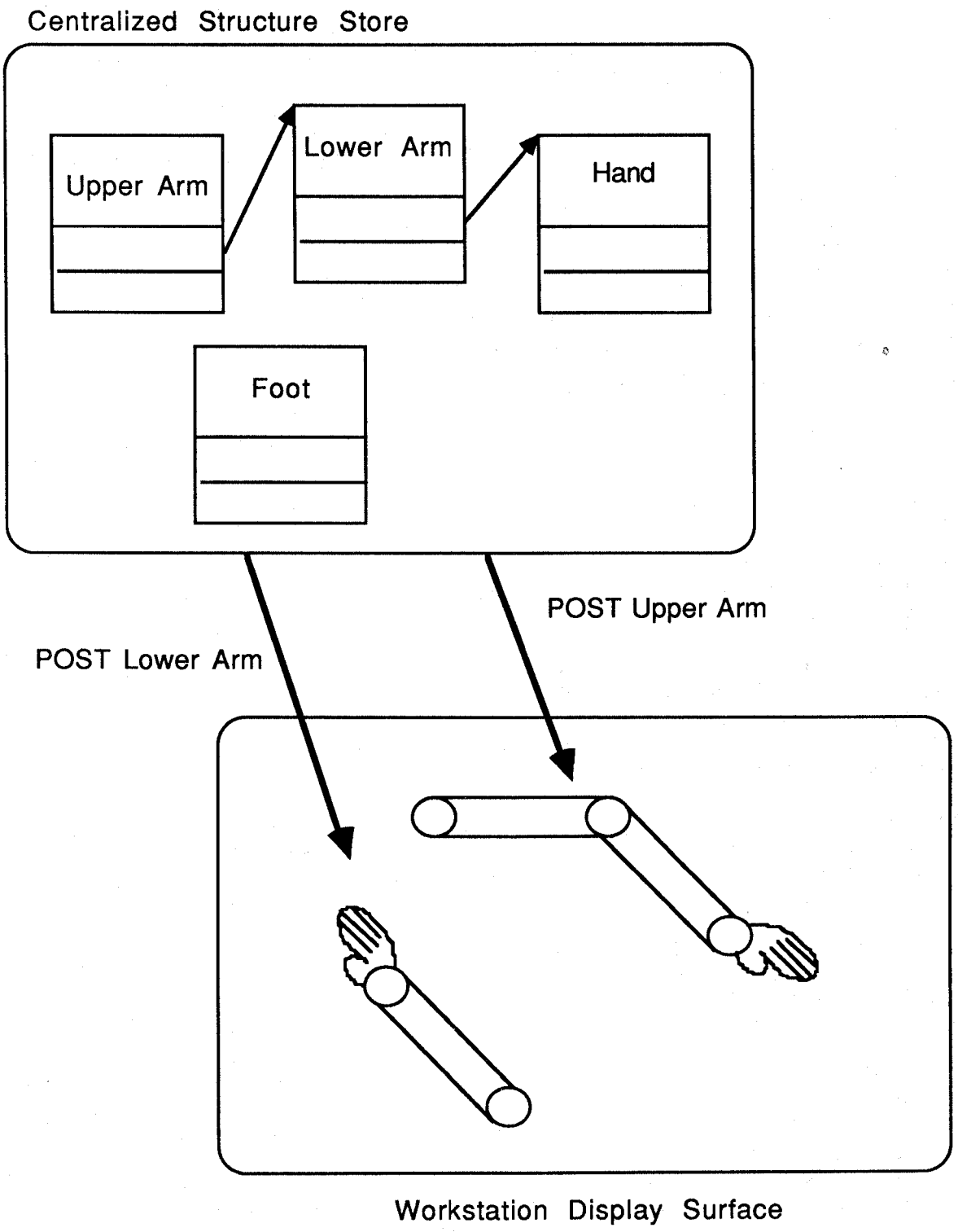


Figure 3 - A structure network with two posted structure networks

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

When a structure is traversed each structure element is interpreted. The initial values in the PHIGS traversal state list are taken from the PHIGS description table. The structure elements of the structure network are executed sequentially. The effect of interpreting a structure element depends on the type of the structure element. Output primitive elements result in the creation of output primitives; structure elements of other types either modify values in the PHIGS traversal state list, or are ignored. Traversal of generalized structure elements may affect all the state lists defined in PHIGS which includes the centralized structure store. When an "execute structure" element is encountered during the traversal, the following actions occur:

a) traversal of the current structure is suspended;

b) the current state of the PHIGS traversal state list is saved;

c) global and local modelling transformations are set (see 4.7.2);

d) the executed structure network is completely traversed;

e) the saved PHIGS traversal state list values are restored;

f) traversal of the current structure is resumed.

A set of default actions is defined so that errors do not occur at traversal time.

4.4.4 Structure editing

PHIGS provides the ability to individually access and modify each structure element within a structure. Editing functions are available to insert new structure elements, replace elements with new structure elements, delete structure elements, navigate within a structure and inquire structure element content.

A structure consists of a sequence of elements. Element positions are implicitly numbered 0..n, where element position 1 is the first element and element position 0 is before the first element. This implicit numbering is always maintained in a consecutive, ascending order with one number per structure element; this means that a structure element position is not fixed and may change as elements are inserted into or deleted from the structure.

A structure is identified for editing by the OPEN STRUCTURE function. If the structure does not exist in the CSS, a structure will be created. Only one structure may be open for editing at any one time. The INQUIRE OPEN STRUCTURE function is provided to determine the identifier of the current open structure. Editing of a structure ends when the CLOSE STRUCTURE function is invoked.

When a structure is opened for editing an *'element pointer'* is established which points at the last element in the structure. The element position may be in the range 0..n, where 0 refers to the beginning of the structure (i.e. before the first element) and n refers to the last element in the structure. An INQUIRE ELEMENT POINTER function is provided to determine the value of the 'element pointer'.

Several functions are provided for manipulating the 'element pointer'. The 'element pointer' may be set to an absolute position with the function SET ELEMENT POINTER, or relative to a position with the function OFFSET ELEMENT POINTER. The offset value may be negative or positive for element pointer movement backwards or forwards respectively. Attempted movement beyond the structure's bounds will always leave the 'element pointer' at the bound it would have exceeded. The 'element pointer' may also be positioned using the function SET ELEMENT POINTER AT LABEL, which positions the 'element pointer' at a "label" structure element. A "label" element is inserted into the structure using the LABEL function. Labels need not be unique within a structure.

When a structure is open the application program may insert additional elements into the structure or

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

replace existing structure elements using the functions which define structure elements (e.g. TEXT, SET POLYLINE INDEX, EXECUTE STRUCTURE). The *'edit mode'*, as defined by the function SET EDIT MODE, defines whether a new element replaces the element pointed to by the 'element pointer' or is added after the element pointed to by the 'element pointer'. In either case the 'element pointer' is then updated to point to the new element.

The COPY ALL ELEMENTS FROM STRUCTURE function can be used to copy all the elements of a structure into the open structure. This copy operation always inserts the copied elements; 'edit mode' does not apply to the copy operation. The 'element pointer' is left pointing at the last element copied.

Three functions are provided to delete elements. The DELETE ELEMENT function will delete the element at which the 'element pointer' is pointing. Consecutive elements may be deleted as a group with the DELETE ELEMENT RANGE function. This function deletes all the elements between and including the lower element position and the higher element position. To delete a group of elements delimited by labels, the DELETE ELEMENTS BETWEEN LABELS function is used. This function will delete all elements between the specified labels. The label elements themselves are not deleted. Following all three delete functions, the 'element pointer' is positioned at the element immediately preceding the element, or group of elements, which was deleted.

All the elements of a structure may be deleted by invoking the EMPTY STRUCTURE function. This function only deletes the contents of the structure and has no effect on references to it. Upon completion of this function, an empty structure will always exist.

Inquiry functions are provided which allow interrogation of the element pointed to by the 'element pointer'. The function INQUIRE CURRENT ELEMENT TYPE AND SIZE returns the type of the element and its size. The parameters of the element may then be obtained by using the INQUIRE CURRENT ELEMENT CONTENT function.

4.4.5 Manipulation of structures in CSS

The following operations apply to structures as whole units in the CSS.

A structure may be deleted by invoking the DELETE STRUCTURE function. This function deletes the structure and all references to it contained in other structures. After the DELETE STRUCTURE call, the structure no longer exists in the CSS. If the deleted structure is referenced by an element in the currently open structure, and if the 'element pointer' is positioned after the "execute structure" element to be deleted, the 'element pointer' is updated such that it still refers to the same element. If the 'element pointer' is positioned at the "execute structure "element to be deleted, the 'element pointer' will be positioned at the preceding element after the structure deletion operation. The DELETE ALL STRUCTURES function is provided to delete all structures from the CSS.

A function is provided to delete a structure network. The DELETE STRUCTURE NETWORK function will delete the indicated structure and structures in the structure network originating with the specified structure. The handling of structures referenced by other structures outside the specified structure network is controlled by a reference handling flag. When this flag is specified as KEEP, those structures referenced outside the specified structure network are not deleted. When this flag is specified as DELETE each structure of the specified structure network is deleted as if DELETE STRUCTURE were called to delete each structure individually. The specified structure is always deleted.

A structure which is to be deleted may currently be posted for display. In this case, the effect is as though the structure were first unposted and then deleted.

Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

A structure's identifier can be changed to another value with the CHANGE STRUCTURE IDENTIFIER function. All the structure identifiers of a given value associated with "execute structure" elements can be modified with the CHANGE STRUCTURE REFERENCES function. Both of these actions can be performed simultaneously by calling the CHANGE STRUCTURE IDENTIFIER AND REFERENCES function. All of these functions can change the arrangement of an existing structure network. If any part of a posted structure network is modified by an invocation of any of the above functions visual changes will result subject to display update state. The functions will create new, empty structures if the specified structures are non-existent.

Inquiry functions are provided to retrieve information about structure identifiers. All existing structure identifiers are returned by the INQUIRE STRUCTURE IDENTIFIERS function.

4.4.6 CSS search and inquiry

PHIGS provides functionality for interrogating the CSS by means of searching and inquiry functions. These may be performed independently of whether a structure is open for editing.

INQUIRE ELEMENT TYPE AND SIZE and INQUIRE ELEMENT CONTENT allow full details of a specified structure element to be determined.

The relationship of a structure to other structures within the CSS may be determined by INQUIRE PATHS TO ANCESTORS and INQUIRE PATHS TO DESCENDANTS. The former returns paths which reference the specified structure while the latter returns paths which are referenced by a particular structure. Paths are returned as a list of element references. The path order and path depth can be specified to determine the portion of each path returned. Path depth determines the maximum number of element references returned in any one path or portion of path. In case of truncation the path order determines whether the head or tail portion of the path is returned. The truncation may result in two or more portions of paths having the same set of element references. Only one such portion will be returned such that all of the returned path portions are distinct.

The ELEMENT SEARCH function is provided to permit searching within a single structure for an element of a particular element type or one of a set of element types. The range of element types to be searched is restricted by the set of element types indicated by element inclusion and exclusion sets. The search is conducted within a single specified structure. The direction of the search from the specified structure element can be specified.

The INCREMENTAL SPATIAL SEARCH (ISS) function allows a structure network to be searched for the next occurrence of a graphical output structure element which satisfies the specified search criteria. The initial starting position for the search and the found position of the structure element that satisfies the search criteria are defined by search paths.

A search path is an *element reference list* which consists of a list of (structure identifier, element position) pairs. The first pair defines the topmost structure of the structure network and the position in that structure of the "execute structure" element that contains the name of the structure identifier that contains the second pair of the list and so on. The last pair in the list defines the structure and position within that structure of the search path position.

The search begins at the structure element following the element identified by the search path position. The search is conceptually a traversal with structure elements being examined sequentially and matched against the search criteria. Element position zero is permitted so that the search may start at the first element of a structure. Search filters are applied to the NAME SET attribute of each graphical output

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

structure element (see 4.5.15). Graphical output structure elements which satisfy the incremental spatial search filters are checked for proximity to the search reference point. The search can either be conducted with modelling clipping applied or not.

For "text" elements the spatial extent is the enclosing rectangle calculated using the values of the geometric attributes (character height, character up vector, text path and text alignment) arising from traversal together with the following values for the workstation dependent attributes: text font, 1; text precision, STROKE; character expansion factor, 1; and character spacing, 0.

For "annotation text relative" elements, the only proximity relationship used is closeness to the annotation reference point.

The search terminates when a graphical output structure element which meets the above criteria is identified during the traversal. The complete search path is returned as the found path. The found path may be used as a search path to start a subsequent search from the position reached.

The search ceiling is an integer which defines a position in the list of the search path. The structure identifier at this position of the list defines a ceiling for the search. Searching will continue until either a graphical output structure element matches the search criteria or the end of the structure identified by the search ceiling is reached.

If a search is successful, the complete found path is returned. An unsuccessful search returns a null found path.

The function is *incremental* in that, having found a match, the search may be continued by invoking INCREMENTAL SPATIAL SEARCH again with the found path as the next starting path. The ceiling may be left unchanged. The search continues from the element position following the last matched position. This allows all elements matching the search criteria to be found for a given portion of a structure network.

Filters may be specified which control which output primitives structure elements in the structures searched are considered.

The filters operate upon the NAME SET associated with an output primitive structure element at traversal time; the NAME SET value that would be defined in the 'current name set' in the PHIGS traversal state list during traversal for display. Each filter consists of an inclusion set and an exclusion set. To be accepted by a filter, the NAME SET has to have at least one name in common with the inclusion set and no names in common with the exclusion set. This is exactly the same condition as is used by the workstation filters which control the highlighting and visibility of primitives.

The filters are organized into two lists, the normal filter list and the inverted filter list which operate in opposite senses. Their effect can be described in the following way.

A structure element is said to be *accepted* if it is declared eligible (see 4.5.15) when the NAME SET is applied to the filter. A structure element is said to be *rejected* if it is declared ineligible when the NAME SET is applied to the filter. For a structure element to be considered a candidate for the spatial search it has to have been accepted by each of the filters in the normal filter list and has to have been rejected by each of the filters in the inverted filter list.

As an example, suppose it is required that only the structure elements corresponding to the pickable (of some pick device) and visible primitives on some workstation take part in the search. Then the normal filter list would contain one entry, which is a copy of the 'pick filter' of the pick device, and the inverted

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

filter list would contain one entry, which is a copy of the 'invisibility filter' of the same workstation. The structure elements whose associated NAME SETs are accepted by the normal filter list correspond to the pickable primitives on that workstation and the structure elements whose associated NAME SETs are rejected by the inverted filter list correspond to the visible primitives on that workstation. The result of the filtering process is that the structure elements corresponding to the pickable and visible primitives will be considered.

If the normal filter list is empty, then all structure elements are considered to have satisfied the acceptance criteria for normal filters. If the inverted filter list is empty, all structure elements are considered to have satisfied the rejection criteria for inverted filters.

4.4.7 Structure archival and retrieval

PHIGS provides functionality for accessing archive files. In ISO/IEC 9592-2 the specification and encoding of archive files and their corresponding generation and retrieval processes are explained. In this part of ISO/IEC 9592 the functionality to access these archive files is defined. An archive file is a medium for storing structure definitions. PHIGS functionality provides for archiving (storing) from the centralized structure store to an archive file, retrieving from an archive file to the centralized structure store, or deleting from an archive file. This set of functions may be applied to a list of structures, a list of structure networks, or to all structures defined either in centralized structure store or in an archive file.

OPEN ARCHIVE FILE initiates and CLOSE ARCHIVE FILE terminates access to archive files. Several archive files may be open simultaneously. The 'maximum number of simultaneously open archive files' is defined in the PHIGS description table and can be inquired with INQUIRE PHIGS FACILITIES. Structures are stored in an archive file with ARCHIVE STRUCTURES, ARCHIVE STRUCTURE NETWORKS, and ARCHIVE ALL STRUCTURES. Structures are recovered from an archive file with RETRIEVE STRUCTURES, RETRIEVE STRUCTURE NETWORKS, and RETRIEVE ALL STRUCTURES. Structures are deleted from an archive file with DELETE STRUCTURES FROM ARCHIVE, DELETE STRUCTURE NETWORKS FROM ARCHIVE, and DELETE ALL STRUCTURES FROM ARCHIVE.

In addition to the above functions there are functions to inquire information about the contents of an archive file. RETRIEVE STRUCTURE IDENTIFIERS returns the identifiers of structures in an archive file. INQUIRE ARCHIVE STATE VALUE returns the archive state of PHIGS. A value of AROP indicates that at least one archive file is open. INQUIRE ARCHIVE FILES returns the names of currently open archive files.

The relationship of a structure to other structures within an archive file may be determined by RETRIEVE PATHS TO ANCESTORS and RETRIEVE PATHS TO DESCENDANTS. The former returns paths which reference the specified structure while the latter returns paths which are referred by a particular structure. Paths are returned as a list of element references. The path order and path depth can be specified to determine the portion of each path returned. Path depth determines the maximum number of element references returned in any one path or portion of path. In case of truncation the path order determines whether the head or tail portion of the path is returned. The truncation may result in two or more portions of paths having the same set of element references. Only one such portion will be returned such that all of the returned path portions are distinct.

As a data protection measure the PHIGS state list contains conflict resolution flags for archival and retrieval. These flags may be set with SET CONFLICT RESOLUTION and inquired with INQUIRE CONFLICT RESOLUTION. The 'archival conflict resolution flag' is used to control what happens when an attempt is made to archive structures whose identifiers already exist in the archive file. The 'retrieval

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

conflict resolution flag' is used to control what happens when an attempt is made to retrieve structures whose identifiers already exist in the CSS. The conflict resolution flags may have the values MAINTAIN, ABANDON or UPDATE. A value of MAINTAIN causes the archival or retrieval operation to only write structures into the archive or CSS if they do not already exist, thereby maintaining existing structures. A value of ABANDON causes the archival or retrieval operation to be abandoned in the event of such conflict, leaving the CSS or archive as it was before the operation started, and an error is generated. A value of UPDATE allows structures to be overwritten.

To permit detection of such conflicts or to determine which structures caused a naming conflict there are functions to inquire potential conflicts. INQUIRE ALL CONFLICTING STRUCTURES compares all structure identifiers in the CSS with those in the specified archive file and returns a list of those which occur in both. INQUIRE CONFLICTING STRUCTURES IN NETWORK compares only those structure identifiers in the specified network in the CSS or the specified archive file with all the structures in the other and returns a list of those which occur in both.

4.4.8 Generalized Structure Elements (GSE)

PHIGS provides the GENERALIZED STRUCTURE ELEMENT to enable access during traversal to special control and attribute related actions that are beyond the scope of the standardized PHIGS structure elements. GSEs are characterized by an identifier and a data record. Traversal of a GSE does not generate an output primitive; GENERALIZED DRAWING PRIMITIVE elements are provided for this purpose. Traversal of GSE's may affect all the state lists defined in PHIGS which includes the CSS. GSE's do not generate errors during traversal.

4.4.9 Application data

An application programmer may insert application specific data into the open structure with the APPLICATION DATA function. This information is for use solely by the application program, and is not interpreted or modified in any way by the PHIGS implementation. During traversal of the structure, the application data is ignored for OUT and OUTIN workstations, but may generate data for MO workstations.

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

4.5 Graphical output

4.5.1 Structure elements and output primitives

The graphical information that is generated by PHIGS and routed to workstations is built up of basic pieces of information called output primitives. Output primitives are generated from structure elements by structure traversal:

a) POLYLINE: PHIGS generates a set of connected lines defined by a point sequence.

b) POLYMARKER: PHIGS generates symbols of one type centred at given positions.

c) TEXT: PHIGS generates a character string at a given position on an arbitrary plane in modelling coordinate space.

d) ANNOTATION TEXT RELATIVE: PHIGS generates a character string at a specified position in an x-y plane in NPC space according the annotation style. It is related to a specified reference point whose coordinates are given in modelling coordinates. The position of the annotation text, the annotation point, is given by an offset specified in NPC space from the transformed reference point.

e) FILL AREA: PHIGS generates a single polygonal area which may be hollow or filled with a uniform colour, a pattern, or a hatch style. No edge control is provided.

f) FILL AREA SET: PHIGS generates a set of polygonal areas which may be hollow or filled with a uniform colour, a pattern, or a hatch style, with or without edges. This allows for specifying areas with holes or disjoint regions that shall be treated as a single entity, particularly for hidden line / hidden surface removal computation.

g) CELL ARRAY: PHIGS generates a two dimensional array of cells with individual colours.

h) GENERALIZED DRAWING PRIMITIVE(GDP): PHIGS addresses special geometric output capabilities of a workstation such as the drawing of spline curves, circular arcs, and elliptic arcs. The objects are characterized by an identifier, a set of points and additional data. PHIGS applies all transformations to the points but leaves the interpretation to the workstation.

Text, fill area, fill area set, and cell array are planar primitives in arbitrary planes. Conceptually these planes have two sides and zero thickness. The viewing transformation determines which side of these primitives is seen. Annotation text relative is also a planar primitive, but is constrained to lie in an x-y plane in NPC space.

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

The plane in which the text primitive lies is defined by a point and two vectors that are contained in the definition of the text primitive. The plane in which the annotation text relative primitive lies is the x-y plane of the normalized projection coordinate system to which the modelling coordinate point is transformed.

The character string used by the text and annotation text primitives contains a sequence of character codes. These may include character codes which result in switching character sets as described in ISO 2022. The default character set is defined in ISO 646 (or a character set including ISO 646). It is implementation dependent whether additional character sets are supported.

It is implementation dependent whether character control codes in the current character set are interpreted. Character codes undefined in the current character set are treated as null characters.

For fill area and fill area set primitives, the plane of the primitives is defined by the points of the respective primitives.

Ensuring that the points of fill areas and fill area sets are coplanar is the responsibility of the application program. While slightly non-coplanar points (e.g. owing to numerical roundoff errors) should be mapped to a plane, the effect of large divergences is implementation dependent.

The interior of a fill area and fill area set is defined in the following way (see figure 4). For a given point, create a straight line starting at that point and going to infinity. If the number of intersections between the straight line and the fill area boundary is odd, the point is within the fill area, otherwise it is outside. If the straight line passes through a fill area vertex tangentially, the intersection count is not affected.

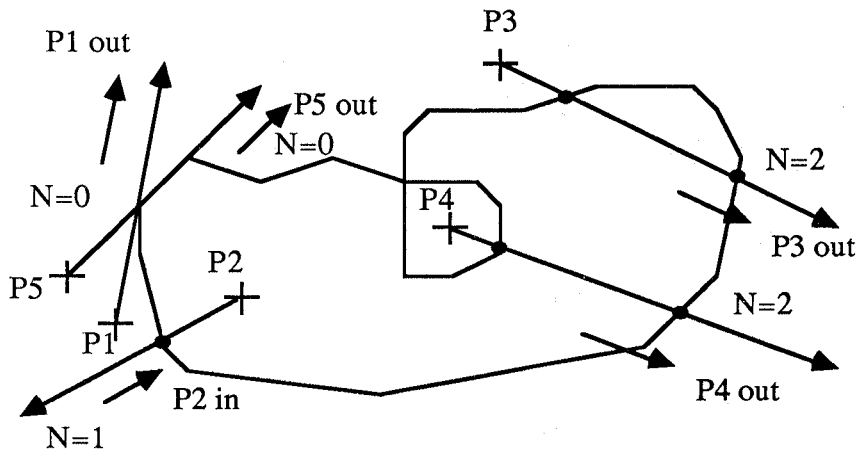


Figure 4 - Area inside of a polygon

If a fill area or fill area set is clipped, the new boundaries generated become part of the displayed interior. While it is recommended that these new boundaries be included in the HOLLOW representation, it is dependent on the hidden line / hidden surface removal method whether or not they are displayed. These boundaries are not displayed as edges for fill area set primitives. Multiple subareas may be generated (see figure 5).

AVA Documents 35
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

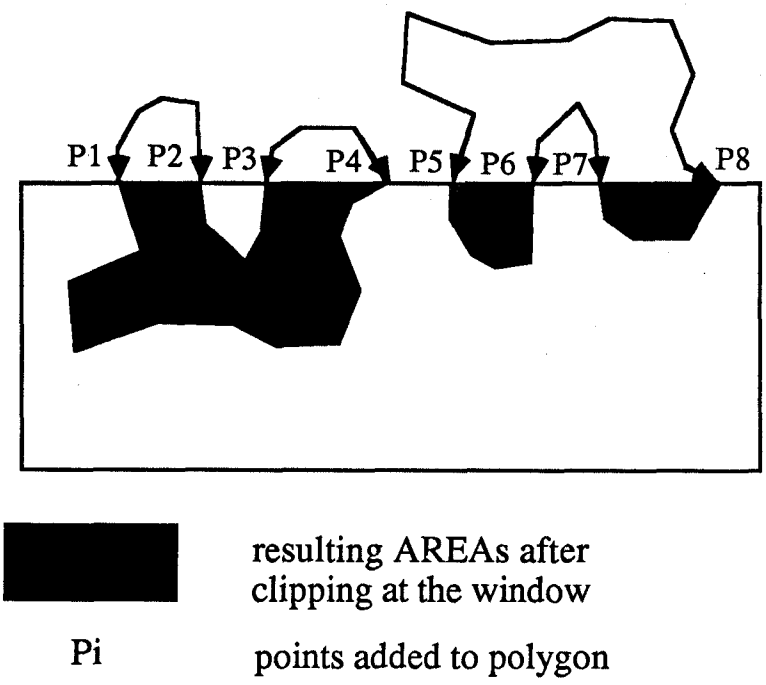


Figure 5 - Examples of FILL AREA and FILL AREA SET clipping

The plane in which the 3D form of the cell array primitive lies is defined by three points, P,Q, and R, contained in the definition of the cell array primitive.

A parallelogram is defined by the points P, Q, R, and (QX+RX-PX, QY+RY-PY, QZ+RZ-PZ). This parallelogram is conceptually divided into a grid of DX×DY cells, where DX and DY are the dimensions of the colour index array. If (PX,PY,PZ) are the coordinates of the cornerpoint P, (QX,QY,QZ) are the coordinates of the cornerpoint Q, and (RX,RY,RZ) are the coordinates of the cornerpoint R, the cell dimensions may be determined as follows:

Each cell has a width defined by:

$$\frac{\sqrt{(PX-QX)^2+(PY-QY)^2+(PZ-QZ)^2}}{DX}$$

Each cell has a height defined by:

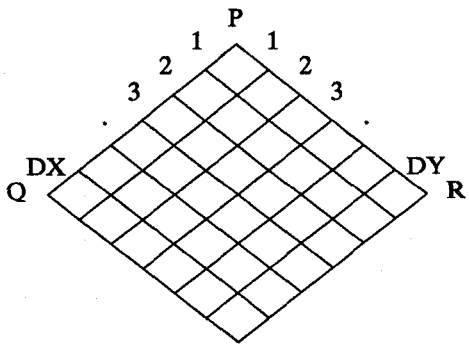
$$\frac{\sqrt{(PX-RX)^2+(PY-RY)^2+(PZ-RZ)^2}}{DY}$$

The colour of each cell is specified by the index of the corresponding element of the colour index array. The colour indices are mapped from the two dimensional cell array on a row-wise basis starting at corner P and proceeding to corner Q and so on. If an index is not present in the colour table on a workstation, index 1 is used on that workstation. Note that this rule also applies to index values less than 0.

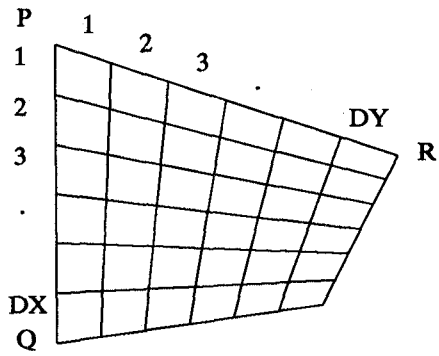
The grid defined by P, Q, R, DX, and DY is subject to all transformations potentially transforming the cells into quadrilaterals. Portions of a transformed cell lying outside the window are clipped.

AVA Documents
Document Sharing Hub

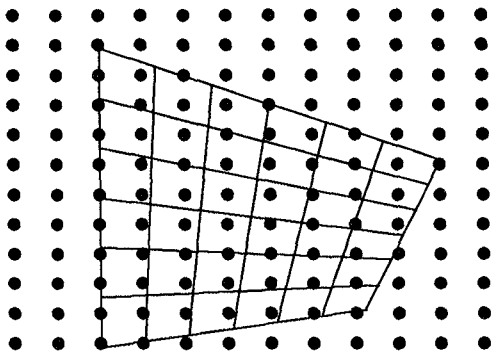
Full document details: ISO_IEC 9592-1_1989



CELL ARRAY
DX × DY CELLS



TRANSFORMED
CELL ARRAY



PIXEL LOCATIONS
ON DISPLAY SURFACE

Cells are mapped onto display surface.
If display location is within cell,then
cell colour is assigned to pixel

Figure 6 - Mapping of CELL ARRAY 3

The 2D form of the cell array primitive is defined by two points P and Q, defining a rectangle, which is taken to be aligned with the modelling coordinate axes. This rectangle is conceptually divided into a grid of DX×DY cells. Each cell has a width of |PX-QX|/DX, and a height of |PY-QY|/DY, where (PX,PY) are the coordinates of the cornerpoint P, and (QX,QY) are the coordinates of the cornerpoint Q. The colour of each cell is specified by the index of the corresponding element of the colour index array. The colour indices are mapped into the two dimensional cell array on a row-wise basis starting at corner (PX,PY) and proceeding to corner Q and so on. If an index is not present in the colour table on a workstation, index 1 is used on that workstation. Note that this rule also applies to index values less than 0.

The rectangular grid defined by P, Q, DX, and DY is subject to all transformations, potentially

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

transforming the rectangular cells into quadrilaterals. Portions of a transformed cell lying outside the window are clipped. Mapping the transformed cells onto the pixels of a raster display is performed by the following rules:

i) If the centrepoint of a pixel lies within the quadrilateral defined by the transformed rectangle, its colour is set.

j) The pixel will be assigned the colour of the cell which contains the pixel's centrepoint. Thus, the pixel colour is selected by point sampling the transformed rectangle at the pixel centrepoint, not by area sampling or filtering. If the pixel centre lies on a cell boundary it is implementation dependent which of the adjacent cells is used to determine the colour of the pixel.

4.5.2 Output primitive attributes

Each output primitive potentially has four types of attributes (geometric, non-geometric, viewing and identification). The first two attribute types determine the exact appearance of the output primitive. The third attribute type selects the viewing and HLHSR parameters for the workstation dependent sections of the pipeline. The fourth attribute type determines visibility, pickability and highlighting of primitives, and is used to identify a primitive, or group of primitives, when a primitive is picked.

The PHIGS primitive attribute specification functions create structure elements in the centralized structure store. During traversal these attribute setting structure elements set corresponding elements of the PHIGS traversal state list. The PHIGS traversal state list is not accessible from the application program. A separate function is provided for each primitive attribute (except the ASFs: see later in this sub-clause) to allow the application program to specify the value of an attribute without unnecessarily specifying the values of other attributes. During creation of an output primitive, during structure traversal, these values are bound to the primitive and cannot be changed afterwards.

Attributes of the first type control the geometric aspects of primitives. These are aspects that affect the shape or size of the entire primitive (for example, CHARACTER HEIGHT for TEXT). Hence they are sometimes referred to as geometric attributes. Geometric attributes are workstation independent and, if they represent coordinate data (points or displacements), they are expressed in modelling coordinates. The geometric attributes of TEXT and ANNOTATION TEXT RELATIVE are an exception as they are specified in a text local coordinate system.

When the values of geometric attributes defined in the PHIGS traversal state list are bound to their respective primitives, the values are subject to the same transformations as the geometric data contained in the definition of the primitive. When the values of annotation text geometric attributes defined in the PHIGS traversal state list are bound to their respective primitives, the values are subject only to the transformations which occur after the transformation into NPC space. Hence changes in the modelling and viewing transformations will not affect the values of geometric attributes already stored in the PHIGS traversal state list.

Attributes of the second type control the non-geometric aspects of primitives. These are aspects that affect a primitive's appearance (for example, linetype for POLYLINE, or colour index for all primitives except CELL ARRAY) or the shape or size of the component parts of the primitive (for example, marker size scale factor for POLYMARKER). Non-geometric aspects do not represent coordinate data. The non-geometric aspects of a primitive may be specified either via a bundle or individually.

Bundled aspects are selected by a bundle index into a bundle table, each entry of which contains non-geometric aspects of a primitive. The non-geometric aspects are workstation dependent in that each

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

workstation has its own set of bundle tables (stored in the workstation state list). The values in a particular bundle (or entry in the bundle table) may be different for different workstations.

For individual specification of aspects, there is a separate attribute for each non-geometric aspect. As with the attributes controlling the geometric aspects, these attributes are workstation independent and are stored during traversal in the PHIGS traversal state list.

For a given non-geometric aspect, the values that can be assigned to the appropriate bundle component are the same as the values that can be assigned to the corresponding attribute for individual specification. Because the bundles are set separately for each workstation, the values of their components are restricted to the valid values for that workstation. For individual attribute specification, such restrictions are not imposed. Default actions for the display of a primitive are defined to occur even if it is created with a value of an individually specified attribute that is invalid on a particular workstation. Thus traversal time errors do not occur.

As described above, GENERALIZED DRAWING PRIMITIVE (GDP) and CELL ARRAY do not have associated bundle tables nor corresponding individually specified attributes. The GDP may use the most appropriate bundle tables or sets of individually specified attributes for each GDP function. For example, if a GDP function is most similar to a FILL AREA, then the fill area bundle table or the set of individually specified fill area attributes would be used. CELL ARRAY contains colour index information as part of its definition but has no other non-geometric aspects.

The method of specification of the non-geometric aspects of a primitive may be chosen separately for each aspect. A further group of primitive attributes, called ASPECT SOURCE FLAGS (ASFs), take the values INDIVIDUAL and BUNDLED to specify the choice. As with the other primitive attributes, these attributes are workstation independent and are stored, during traversal, in the PHIGS traversal state list. There is one ASF for each non-geometric aspect. The initial values of all of the ASFs are the same; this may be either BUNDLED or INDIVIDUAL, the choice being implementation dependent. If the initial values are not altered the system will operate

a) as if individual specification of non-geometric aspects were not a system feature, if the initial values of all of the ASFs are BUNDLED;

b) as if specification of non-geometric aspects via a bundle were not a system feature, if the initial values of all of the ASFs are INDIVIDUAL.

The flags may be set during structure traversal by the structure element "set individual ASF", which is generated from the SET INDIVIDUAL ASF function.

When a primitive is displayed, the values of the non-geometric aspects with which it is displayed are determined as follows:

c) if the ASF for an aspect is INDIVIDUAL, the value used on all workstations is the value of the corresponding individually specified attribute of the primitive;

d) if the ASF for an aspect is BUNDLED, the value used on a workstation is the value of the corresponding component of the workstation dependent bundle, pointed to by the bundle index.

Colour is a non-geometric aspect of a primitive, which is specified as an index into a separate colour table. There is a colour table on each workstation into which all the colour indices point. Similarly, other entries in a bundle, or corresponding individually specified attributes, may be indices either into another workstation table (for example, style index when interior style PATTERN is used) or into a workstation dependent list (for example, linetypes for polyline).

Documents Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

There are two attributes of the third type, the VIEW INDEX and HLHSR IDENTIFIER. The attribute VIEW INDEX identifies an entry in a workstation dependent view table. The attribute HLHSR IDENTIFIER supplies HLHSR information to the workstation. The interpretation of the identifier is implementation dependent (see 4.5.14). Implementation of the functions to control hidden line / hidden surface removal is obligatory, but the functions may have no effect on the appearance of primitives.

There are two attributes of the fourth type, which apply to each primitive, namely PICK IDENTIFIER and NAME SET. The attribute PICK IDENTIFIER is used for identifying a primitive, or a group of primitives, in a structure when that structure is picked. The attribute NAME SET is a set of classifications defining, in part, the eligibility of the primitive for highlighting, invisibility, pickability and incremental spatial search.

The explicitly specified attributes that apply to each output primitive (attributes controlling non-geometric aspects, geometric attributes, view attributes and pick identifier) are:

e) POLYLINE:
- POLYLINE INDEX
- LINETYPE
- LINEWIDTH SCALE FACTOR
- POLYLINE COLOUR INDEX
- LINETYPE ASF
- LINEWIDTH SCALE FACTOR ASF
- POLYLINE COLOUR INDEX ASF
- VIEW INDEX
- HLHSR IDENTIFIER
- PICK IDENTIFIER
- NAME SET

f) POLYMARKER:
- POLYMARKER INDEX
- MARKER TYPE
- MARKER SIZE SCALE FACTOR
- POLYMARKER COLOUR INDEX
- MARKER TYPE ASF
- MARKER SIZE FACTOR ASF
- POLYMARKER COLOUR INDEX ASF
- VIEW INDEX
- HLHSR IDENTIFIER
- PICK IDENTIFIER
- NAME SET

g) TEXT:
- TEXT INDEX
- TEXT FONT
- TEXT PRECISION
- CHARACTER EXPANSION FACTOR
- CHARACTER SPACING
- TEXT COLOUR INDEX
- TEXT FONT ASF
- TEXT PRECISION ASF
- CHARACTER EXPANSION FACTOR ASF
- CHARACTER SPACING ASF
- TEXT COLOUR INDEX ASF
- CHARACTER HEIGHT

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

CHARACTER UP VECTOR
TEXT PATH
TEXT ALIGNMENT
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER
NAME SET

h) ANNOTATION TEXT RELATIVE:

TEXT INDEX
TEXT FONT
TEXT PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR INDEX
TEXT FONT ASF
TEXT PRECISION ASF
CHARACTER EXPANSION FACTOR ASF
CHARACTER SPACING ASF
TEXT COLOUR INDEX ASF
ANNOTATION TEXT CHARACTER HEIGHT
ANNOTATION TEXT CHARACTER UP VECTOR
ANNOTATION TEXT PATH
ANNOTATION TEXT ALIGNMENT
ANNOTATION STYLE
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER
NAME SET

i) FILL AREA:

INTERIOR INDEX
INTERIOR STYLE
INTERIOR STYLE INDEX
INTERIOR COLOUR INDEX
INTERIOR STYLE ASF
INTERIOR STYLE INDEX ASF
INTERIOR COLOUR INDEX ASF
PATTERN SIZE
PATTERN REFERENCE POINT
PATTERN REFERENCE VECTORS
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER
NAME SET

j) FILL AREA SET:

INTERIOR INDEX
INTERIOR STYLE
INTERIOR STYLE INDEX
INTERIOR COLOUR INDEX
INTERIOR STYLE ASF
INTERIOR STYLE INDEX ASF
INTERIOR COLOUR INDEX ASF
EDGE INDEX

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

		EDGE FLAG EDGETYPE EDGEWIDTH SCALE FACTOR EDGE COLOUR INDEX EDGE FLAG ASF EDGETYPE ASF EDGEWIDTH SCALE FACTOR ASF EDGE COLOUR INDEX ASF PATTERN SIZE PATTERN REFERENCE POINT PATTERN REFERENCE VECTORS VIEW INDEX HLHSR IDENTIFIER PICK IDENTIFIER NAME SET
k)	CELL ARRAY:	VIEW INDEX HLHSR IDENTIFIER PICK IDENTIFIER NAME SET
l)	GENERALIZED DRAWING PRIMITIVE:	Zero or more of the sets e) to k) except that VIEW INDEX, HLHSR IDENTIFIER, PICK IDENTIFIER, and NAME SET are always attributes.

The attributes for each primitive are described in 4.5.3 to 4.5.15. VIEW INDEX is described in more detail in 4.5.13. In the descriptions, attributes appear in upper case (for example, the attributes CHARACTER HEIGHT and PICK IDENTIFIER); aspects appear in both upper and lower case, according to their context. Geometric aspects are always controlled by geometric attributes and so appear in upper case (for example, the aspect CHARACTER HEIGHT). Non-geometric aspects may be controlled via a <bundle> INDEX or by individually specified attributes. Non-geometric aspects appear in lower case unless the corresponding individually specified attributes are being used. Individually specified attributes appear in upper case (for example, the aspect linetype but the individually specified LINETYPE).

The entries in the bundle, pattern and colour tables may be set separately for each workstation. Some predefined definitions for table entries are contained in the workstation description table and are used as initial values. The application program may select a standard definition or may define the values of a specific entry explicitly. Only the most commonly used (or anticipated) combinations of values need be predefined for each output type workstation. At least those predefined entries with indices up to the minimum number of predefined entries are distinguishable from each other. Other combinations of values can be specified by the SET <bundle | VIEW | PATTERN | COLOUR> REPRESENTATION function, possibly after inquiring the workstation capabilities. The tables, which are on every workstation of categories OUTPUT and OUTIN (i.e. they are workstation attributes), are:

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

view table
polyline bundle table
polymarker bundle table
text bundle table
interior bundle table
edge bundle table
pattern table
colour table

The values in these tables may be changed dynamically. The only way of changing the aspects of a primitive that are stored in a bundle table is by changing the bundle table entry. However, note that a change in a bundle table entry can be reflected in a displayed primitive only if the values of the corresponding primitive's ASFs for the aspects in the bundle table are BUNDLED. The entry 'dynamic modification accepted' in the workstation description table indicates which changes lead to an implicit regeneration (may be deferred) (IRG), and which can be performed immediately (IMM). The display update states are explained in more detail in 4.6.3.

During traversal, and when initiating a measure process (see 4.8), if an index into a bundle table or individual attribute is out of range or undefined, a default value is used. This applies to the workstation state list tables and to linetype, marker type, text font and edge type. For marker type the default value is three (3). Otherwise, the default value is one (1).

4.5.3 Polyline attributes

Polyline has no geometric attributes. The representation of polyline on the workstation is controlled by the POLYLINE INDEX, or the set of individually specified polyline attributes (LINETYPE, LINEWIDTH SCALE FACTOR and POLYLINE COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for linetype, linewidth scale factor and polyline colour index. The POLYLINE INDEX is a pointer into the polyline bundle table, each entry of which contains values for linetype, linewidth scale factor and polyline colour index.

Linetypes 1 to 4 are predefined as solid, dashed, dotted and dashed-dotted. Every workstation of category OUTPUT or OUTIN realizes linetypes 1 to 4 with recognizable styles. Linetypes greater than 4 are reserved for registration (see 4.1.2). Linetypes less than 1 may be available but their styles are implementation dependent. The linetype specifies a sequence of line segments and gaps which are repeated to draw a polyline. It is workstation dependent whether this sequence is restarted or continued at the start of the polyline, at the start of a clipped piece of polyline, and at each vertex of a polyline.

The linewidth is calculated as a nominal linewidth multiplied by the linewidth scale factor. This value is mapped by the workstation to the nearest available linewidth. For example, if the value is less than or equal to zero (0), the thinnest available linewidth is used.

4.5.4 Polymarker attributes

Polymarker has no geometric attributes. The representation of polymarker at the workstation is controlled by the POLYMARKER INDEX, or the set of individually specified polymarker attributes (MARKER TYPE, MARKER SIZE SCALE FACTOR and POLYMARKER COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for marker type, marker size scale factor, and polymarker colour index. The POLYMARKER INDEX is a pointer into the polymarker

Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989

bundle table, each entry of which contains values for marker type, marker size scale factor and polymarker colour index.

Marker types 1 to 5 are predefined as dot, plus sign, asterisk, circle, and diagonal cross each centred on the positions they are identifying. Every workstation of category OUTPUT or OUTIN realizes marker types 1 to 5 with recognizable shapes at the given positions. Marker types greater than 5 are reserved for registration (see 4.1.2). Marker types less than 1 may be available but their forms are implementation dependent.

The marker size is calculated as a nominal size multiplied by the marker size scale factor. This size is mapped by the workstation to the nearest available size. For example, if the size is less than or equal to zero (0), the smallest available marker size is used. Marker type 1 is always displayed as the smallest displayable dot.

The marker is visible (independent of whether it is obscured by another primitive) if, and only if, the marker position is within the clipping volume. The clipping of markers whose position is inside the clipping volume, but whose rendering would extend outside the clipping volume, is workstation dependent.

Since the polymarker has no geometric attributes, transformations affect only the marker position and not the appearance of the marker glyph.

4.5.5 Text attributes

Text is a planar primitive. The plane of the text primitive is defined by a point and two vectors contained in the definition of the text primitive. The point is called the text position. The two vectors are called the first and second text direction vectors. The point and these vectors define a text plane and a local coordinate system on that plane in which all geometric text attributes are measured; the coordinate system is known as the *text local coordinate* system. The text local coordinate system has unit size unscaled from modelling coordinates.

Let P be the text position and U and V be the first and second text direction vectors respectively. The text plane is the plane that contains P and is perpendicular to $U \times V$. The origin of the text local coordinate system is at the point P. The positive X axis of the text local coordinate system is parallel to U. Let T be a vector perpendicular to U and in the text plane. If $T \cdot V$ is positive, then the positive Y axis is parallel to T; otherwise, the positive Y axis is opposite T. In the following all references to coordinates refer to this text local coordinate system unless explicitly stated otherwise.

Text has the geometric attributes CHARACTER HEIGHT, CHARACTER UP VECTOR, TEXT PATH, and TEXT ALIGNMENT which are specified and used as described in this sub-clause.

Text also has two implicitly specified geometric attributes CHARACTER WIDTH and CHARACTER BASE VECTOR. These are implicitly specified by the functions SET CHARACTER HEIGHT and SET CHARACTER UP VECTOR respectively. They otherwise behave like ordinary text geometric attributes (their values are bound to TEXT primitives when the primitives are created and cannot be changed afterwards and these values are subject to the same transformations as the geometric data contained in the definition of the primitive). The CHARACTER BASE VECTOR is set to a vector, of arbitrary length, at a right angle in the clockwise direction, to the CHARACTER UP VECTOR.

The representation of text at the workstation is controlled by the TEXT INDEX, or the set of individually specified text attributes (TEXT FONT, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, and TEXT COLOUR INDEX) or some combination of the two,

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 9592-1_1989