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)

Contents

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

Annexes

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.

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:

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.

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.

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.

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.

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*.

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*.

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*.

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.