BRITISH STANDARD

BS ISO 8805:1988

Information processing systems — Computer graphics — Graphical Kernel System for Three Dimensions (GKS-3D) functional description

ICS 35.140

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

National foreword

This British Standard reproduces verbatim ISO 8805:1988 and implements it as the UK national standard. It supersedes BS EN 28805:1992 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/31, Computer Graphics and Image Processing, which has the responsibility to:

— aid enquirers to understand the text;

— present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;

— monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled "International Standards Correspondence Index", or by using the "Find" facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Policy and Strategy Committee on 2 November 2001

Summary of pages

This document comprises a front cover, an inside front cover, an ISO title page, pages ii to vii, a blank page, pages 1 to 379 and a back cover.

The BSI copyright date displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

© BSI 2 November 2001

ISBN 0 580 38579 5

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

INTERNATIONAL STANDARD

ISO 8805

First edition
1988-09-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Information processing systems — Computer graphics — Graphical Kernel System for Three Dimensions (GKS-3D) functional description

Systèmes de traitement de l'information — Infographie — Système graphique de base pour trois dimensions (GKS-3D) description fonctionnelle

Reference number
ISO 8805:1988 (E)

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8805 was prepared by Technical Committee ISO/TC 97, *Information processing systems.*

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.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Contents

0 Introduction 1
1 Scope and field of application 4
2 References 5
3 Definitions 6
4 The Graphical Kernel System for Three Dimensions 14
- **4.1** About this International Standard 14
  - **4.1.1** Specification 14
  - **4.1.2** Registration 14
- **4.2** Introduction to GKS-3D 15
- **4.3** Concepts 17
- **4.4** Graphical output 20
  - **4.4.1** Output primitives 20
  - **4.4.2** Output primitive attributes 21
  - **4.4.3** Polyline attributes 26
  - **4.4.4** Polymarker attributes 27
  - **4.4.5** Text attributes 27
  - **4.4.6** Text extent and concatenation 39
  - **4.4.7** Fill area attributes 39
  - **4.4.8** Fill area set attributes 41
  - **4.4.9** Cell array attributes 42
  - **4.4.10** Generalized Drawing Primitive attributes 42
  - **4.4.11** Colour 42
  - **4.4.12** View index 43
  - **4.4.13** Hidden line/hidden surface removal (HLHSR) identifier 43
- **4.5** Workstations 44
  - **4.5.1** Workstation characteristics 44
  - **4.5.2** Selecting a workstation 45
  - **4.5.3** Deferring picture changes 46
  - **4.5.4** Clearing the display space 50
  - **4.5.5** Elimination of primitives outside segments 50
  - **4.5.6** Sending messages to a workstation 51
  - **4.5.7** Hidden line/hidden surface removal 51
- **4.6** Coordinate systems and transformations 52
  - **4.6.1** Normalization transformations 52
  - **4.6.2** Clipping 53
  - **4.6.3** Workstation transformations 53
  - **4.6.4** Transformation of locator input 56
  - **4.6.5** Transformation of stroke input 59
  - **4.6.6** Viewing 59
  - **4.6.7** Viewing utility functions 61
- **4.7** Segments 70
  - **4.7.1** Introduction to segments 70
  - **4.7.2** Segment attributes 71
  - **4.7.3** Segment transformations 72
  - **4.7.4** Clipping and WDSS 72

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

4.7.5 Workstation Independent Segment Storage 72
4.7.6 WISS functions and clipping 73
4.8 Graphical input 74
4.8.1 Introduction to logical input devices 74
4.8.2 Logical input device model 75
4.8.3 Operating modes of logical input devices 76
4.8.4 Measures of each input class 79
4.8.5 Input queue and current event report 80
4.8.6 Initialization of input devices 81
4.8.7 Locator and stroke input using 2D input devices 82
4.9 GKS-3D Metafile interface 83
4.10 GKS-3D levels 85
4.10.1 Introduction to levels 85
4.10.2 The level structure 85
4.10.3 Level functionality 86
4.11 States of GKS-3D and inquiry functions 91
4.11.1 Description of states 91
4.11.2 Inquiry functions 92
4.12 Error handling 94
4.13 Special interfaces between GKS-3D and the application program 96
4.14 2D functions 97
5 GKS-3D functions 101
5.1 Notational conventions 101
5.2 Control functions 102
5.3 Output functions 111
5.4 Output attributes 125
5.4.1 Workstation independent primitive attributes 125
5.4.2 Workstation attributes (representations) 140
5.5 Transformation functions 149
5.5.1 Normalization transformation 149
5.5.2 View transformation 152
5.5.3 Workstation transformation 154
5.6 Segment functions 158
5.6.1 Segment manipulation functions 158
5.6.2 Segment attributes 163
5.7 Input functions 167
5.7.1 Initialization of input devices 167
5.7.2 Setting the mode of input devices 182
5.7.3 Request input functions 185
5.7.4 Sample input functions 190
5.7.5 Event input functions 195
5.8 Metafile functions 200
5.9 Inquiry functions 202
5.9.1 Introduction to inquiry functions 202
5.9.2 Inquiry function for operating state value 202
5.9.3 Inquiry functions for GKS-3D description table 202
5.9.4 Inquiry functions for GKS-3D state list 204
5.9.5 Inquiry functions for workstation state list 214
5.9.6 Inquiry functions for workstation description table 245
5.9.7 Inquiry functions for segment state list 275
5.9.8 Pixel inquiries 276
5.9.9 Inquiry function for GKS-3D error state list 279
5.10 Utility functions 280
5.11 Error handling 284
6 GKS-3D data structures 286
6.1 Notation and data types 286
6.2 Operating state 289

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

6.3 GKS-3D description table 290
6.4 GKS-3D state list 291
6.5 Workstation state list 294
6.6 Workstation description table 298
6.7 Segment state list 303
6.8 GKS-3D error state list 304
A Function lists 305
A.1 Alphabetic 305
A.2 Order of appearance 310
A.2.1 Control functions 5.2 310
A.2.2 Output functions 5.3 310
A.2.3 Output attributes 5.4 310
A.2.3.1 Workstation independent primitive attributes 5.4.1 311
A.2.3.2 Workstation attributes (representations) 5.4.2 311
A.2.4 Transformation functions 5.5 311
A.2.4.1 Normalization transformation 5.5.1 311
A.2.4.2 View transformation 5.5.2 312
A.2.4.3 Workstation transformation 5.5.3 312
A.2.5 Segment functions 5.6 312
A.2.5.1 Segment manipulation functions 5.6.1 312
A.2.5.2 Segment attributes 5.6.2 312
A.2.6 Input functions 5.7 312
A.2.6.1 Initialization of input devices 5.7.1 312
A.2.6.2 Setting the mode of input devices 5.7.2 313
A.2.6.3 Request input functions 5.7.3 313
A.2.6.4 Sample input functions 5.7.4 313
A.2.6.5 Event input functions 5.7.5 313
A.2.7 Metafile functions 5.8 313
A.2.8 Inquiry functions 5.9 314
A.2.8.1 Inquiry function for operating state value 5.9.2 314
A.2.8.2 Inquiry functions for GKS-3D description table 5.9.3 314
A.2.8.3 Inquiry functions for GKS-3D state list 5.9.4 314
A.2.8.4 Inquiry functions for workstation state list 5.9.5 314
A.2.8.5 Inquiry functions for workstation description table 5.9.6 315
A.2.8.6 Inquiry functions for segment state list 5.9.7 316
A.2.8.7 Pixel inquiries 5.9.8 316
A.2.8.8 Inquiry function for GKS-3D error state list 5.9.9 316
A.2.9 Utility functions 5.10 316
A.2.10 Error handling 5.11 316
A.3 Ordered by level 316
A.3.1 Level 0a 316
A.3.2 Level 0b 319
A.3.3 Level 0c 320
A.3.4 Level 1a 320
A.3.5 Level 1b 321
A.3.6 Level 1c 322
A.3.7 Level 2a 322
A.4 Applicability to workstation groups 323
B Error list 329
B.1 Implementation dependent 329
B.2 States 329
B.3 Workstations 329
B.4 Transformations 329
B.5 Output attributes 330
B.6 Output primitives 330
B.7 Segments 331
B.8 Input 331

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

B.9 Metafiles ... 331
B.10 Escape ... 331
B.11 Miscellaneous ... 331
B.12 System ... 331
B.13 3D transformations ... 332
B.14 3D output attributes ... 332
B.15 3D output primitives ... 332
B.16 Reserved errors ... 332
C Interfaces ... 333
C.1 General ... 333
C.2 Language binding ... 333
C.3 Implementation ... 334
D Allowable differences in GKS-3D implementations ... 336
D.1 General ... 336
D.2 Global differences ... 336
D.3 Workstation dependent differences ... 337
E Metafile structure ... 340
E.1 Metafiles ... 340
E.1.1 General ... 340
E.1.2 ISO 8632 Metafile ... 340
E.1.3 Metafile designed for GKS-3D ... 341
E.2 File format and data format ... 341
E.3 Generation of metafiles ... 342
E.4 Interpretation of metafiles ... 346
E.4.1 General ... 346
E.4.2 Control items ... 346
E.4.3 Output primitives ... 346
E.4.4 Output primitive attributes ... 346
E.4.5 Workstation attributes ... 346
E.4.6 Transformations ... 346
E.4.7 Segment manipulation ... 347
E.4.8 Segment attributes ... 347
E.4.9 Items from a GKSM written by a GKS system ... 347
E.5 Control items ... 347
E.6 Items for output primitives ... 350
E.7 Items for output primitive attributes ... 351
E.8 Items for workstation attributes ... 355
E.9 Items for transformations ... 357
E.9.1 Items for clipping ... 357
E.9.2 Items for viewing ... 357
E.9.3 Items for HLHSR ... 357
E.9.4 Items for workstation transformation ... 358
E.10 Items for segment manipulation ... 358
E.11 Items for segment attributes ... 358
E.12 User items ... 359
F Sample programs ... 360
G GKS-3D functions summary ... 369
G.1 Control functions ... 369
G.2 Output functions ... 369
G.3 Output attributes ... 370
G.3.1 Workstation independent primitive attributes ... 370
G.3.2 Workstation attributes (representations) ... 372
G.4 Transformation functions ... 372
G.4.1 Normalization transformation ... 372
G.4.2 Viewing transformation ... 372
G.4.3 Workstation transformation ... 373

AVA Documents
Document Sharing Hub
Full document details: BS ISO 8805:1988

G.5 Segment functions 373
G.5.1 Segment manipulation functions 373
G.5.2 Segment attributes 374
G.6 Input functions 374
G.6.1 Initialization of input devices 374
G.6.2 Setting mode of input devices 375
G.6.3 Request input functions 375
G.6.4 Sample input functions 375
G.6.5 Event input functions 376
G.7 Metafile functions 377
G.8 Inquiry functions 377
G.9 Utility functions 377
G.10 Error handling 377
H Colour models 379

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Documents
Document Sharing Hub

Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) functional description

0 Introduction

The Graphical Kernel System for Three Dimensions (GKS-3D) provides a set of functions for:

a) definition and display of 2D and 3D graphical data;

b) storage and manipulation of graphical data;

c) input of graphically related data.

The main reasons for introducing this International Standard for computer graphics is:

d) to allow application programs involving graphics to be easily portable between different installations;

e) to aid the understanding and use of graphics methods by application programmers;

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

In order to reach these main objectives, the GKS-3D design was based on the following requirements:

g) GKS-3D should include all the capabilities that are essential for the whole spectrum of graphics, from simple passive output to highly interactive applications.

h) The whole range of graphics devices, including vector and raster devices, microfilm recorders, storage tube displays, refresh displays and colour displays should be controllable by GKS-3D in a uniform way.

i) GKS-3D should provide all the capabilities required by a majority of applications.

These requirements were used to formulate a number of principles that were used to judge specific design alternatives. Thus it was possible to contribute to the overall design goals while focussing on certain aspects. Five design aspects were identified, each having a group of principles

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

j) Design goals: The following principles should not be violated by any technical design:

1) consistency: the mandatory requirements of GKS-3D should not be mutually contradictory;

2) compatibility: other standards or commonly accepted rules of practice should not be violated;

3) orthogonality: the functions or modules of GKS-3D should be independent of each other, or the dependency should be structured and well defined.

k) Functional capabilities: The following principles were used to define the extent of GKS-3D:

1) completeness: all functions that a majority of applications want to use on a given level of functionality should be included;

2) minimality: functions that are unnecessary for applications of a given level of functionality should not be provided;

3) compactness: an application should be able to achieve a desired result by a set of functions and parameters that is as small as possible;

4) richness: a rich set of functions offers an extensive range of facilities that stretches beyond the basic functions and includes higher order capabilities.

It is obvious that there is a trade off between the principles in this group. Therefore, the functions of GKS-3D are organized in nine levels. An implementation of GKS-3D provides precisely the functions of one of these levels. While the lowest level contains only a minimal set of functions, higher levels are allowed to extend beyond the basic needs towards greater richness.

l) User interface design: The following principles were used to define the user interface design:

1) user friendliness: GKS-3D should allow the design of a desirable user interface;

2) clarity: the concepts and functional capabilities of GKS-3D should be easily understandable, especially by the application programmer;

3) error handling: failure of system functions or modules, caused by errors of the system itself or by the application program, should be treated in such a way that the error reaction is clearly understandable and informative to the application programmer and that the impact on the system and the application program is as small as possible.

Clarity and sound error handling are essential parts of user friendliness. Error handling is an integral part of GKS-3D. To aid clarity, the system and its state can be presented to the user in an easily comprehensible manner.

Clarity applies not only to the system design but also to the system description. To this end, the GKS-3D specification is divided into a general description, a description of the underlying logical data structures representing the state of the system, and a description of the functions and their effects on these data structures.

m) Graphics devices: The following principles are associated with the range of graphics devices that can be addressed by GKS-3D:

1) device independence: GKS-3D functions should be designed to allow an application program, using these functions, to address facilities of quite different graphics output and input devices without modification of the program structure;

2) device richness: the full capabilities of a wide range of different graphics output and input devices should be accessible from the functions of GKS-3D.

These principles led to a fundamental concept underlying the GKS-3D architecture: the concept of multiple independent graphical workstations connected to and driven by GKS-3D. The application program can inquire the capabilities of every workstation. The GKS-3D design includes escape functions that are easily identifiable within an application program and can be used to access special facilities of a particular device.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Introduction

n) Implementation: The last group of principles is related to the implementation of GKS-3D:

1) implementability: it should be possible to support the GKS-3D functions in most host languages, on most operating systems and with most graphics devices;

2) language independence: it should be possible to access the facilities of GKS-3D from all International Standard programming languages;

3) efficiency: GKS-3D should be capable of being implemented without time consuming algorithms;

4) robustness: the operator and application programmer should be protected in the best possible way from hardware or software failure of the system.

The five groups of principles are interconnected. For example, design goals and functional capabilities both contribute to user friendliness. Efficiency is also important when considering response time in an interactive environment. Some principles may be conflicting, such as richness versus minimality, comprehensive error handling versus efficiency, and compactness versus device richness. Compromises needed to be made to achieve the overall design objective: GKS-3D should have an easily comprehensible structure and a set of functions that enables a vast majority of computer graphics users to design portable, device independent application programs addressing the whole range of computer graphics equipment.

o) Relationship to ISO 7942: This International Standard adheres to the following design goals with respect to ISO 7942:

1) the relationship between the functions in ISO 7942 and the functions described in this International Standard should be precisely defined and not be implementation dependent;

2) the format of functions and their parameters in ISO 7942 will remain unchanged in this International Standard;

3) additional functions will be added solely to support the new 3D functionality.

p) Compatibility of Application Programs: a major goal of this International Standard is that application programs written using the functions specified in ISO 7942 will run without modification on implementations conforming to this International Standard and produce the same results on workstations of the same type, subject to the allowable differences as detailed in Annex D. However, it is recognized that in certain situations, especially involving the use of metafiles, this goal cannot always be achieved (see Annex D for further details).

q) Relationship to ISO 9592: This International Standard has the following relationship to ISO 9592 (PHIGS):

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 provides data structure facilities which can be rendered as graphical output, rather than being purely directed towards providing graphical output and interaction.

GKS/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 almost identical to 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 PHIGS Central Structure Store which generates the primitive picked by the operator.

Annexes C to H are given for information; they do not form part of the specification.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

1 Scope and field of application

This International Standard specifies a set of functions for computer graphics programming, the Graphical Kernel System for Three Dimensions (GKS-3D). GKS-3D is a basic graphics system for applications that produce computer generated three dimensional pictures on graphics output devices. It supports operator input and interaction by supplying basic functions for graphical input and picture segmentation. It allows storage and dynamic modification of pictures. A fundamental concept in GKS-3D is the workstation, consisting of a number of input devices and a single output device. 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. This International Standard includes functions for storage on and retrieval from an external graphics file. Also, the functions are organized in upward compatible levels with increasing capabilities.

This International Standard provides application programs with the capability to define and display 3D graphical primitives, specified using 3D coordinates. In addition, the GKS-3D input model provides 3D locator and stroke input.

The output primitives of this International Standard differ from the output primitives of ISO 7942 in the following ways:

a) the polyline and polymarker primitives are defined by an arbitrary sequence of 3D coordinates;

b) the text, fill area and cell array primitives remain planar, but may lie in a plane positioned arbitrarily in 3D space;

c) the fill area set primitive is introduced.

Functions are provided in this International Standard to generate these primitives.

The functions of ISO 7942 can be used in this International Standard. However, the output functions generate instances of 3D primitives — instances that are constrained to lie in a plane. Furthermore, additional capabilities are provided in this International Standard to:

d) specify views of 3D objects;

e) obtain information from 3D input devices;

f) permit workstations to perform hidden line/hidden surface removal (HLHSR).

This International Standard provides no specific functions for controlling such rendering techniques as light source, shading, texturing, and shadow computations. However, workstations may apply these techniques, locally as the primitives and attributes are realized, provided that the general guidelines of Annex D concerning allowable workstation differences are followed.

NOTE - For certain parameters of the functions, GKS-3D defines value ranges as being reserved for registration (see 4.1.2). The meanings of these values will be defined using the established procedures.

GKS-3D defines a language independent nucleus of a graphics system. For integration into a programming language, GKS-3D is embedded in a language dependent layer obeying the particular conventions of that language.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

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 numerical 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* [1)]

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

ISO/IEC 8806 *Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) Language Bindings* [2)]

- *Part 1 : FORTRAN*
- *Part 2 : PASCAL*
- *Part 3 : ADA*
- *Part 4 : C*

ISO/IEC 9592, *Information processing systems - Computer graphics - The Programmer's Hierarchical Interactive Graphics System (PHIGS)*

- *Part 1 : Functional description.*
- *Part 2 : Archive file format*
- *Part 3 : Clear text encoding.of archive file*

CIE Recommendations on colour space - supplement to CIE publication 15

CIE 1976 Supplementary standard colour metric of server and coordinate systems

1) Two addenda are being produced which support segmentation and three dimensions.
2) At present at the stage of draft; publication anticipated in due course.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

3 Definitions

For the purposes of this International Standard the following definitions apply.

NOTE - As far as possible, commonly accepted graphics terminology is used.

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

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

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

3.4 aspects of primitives: Ways in which the appearance of a primitive can vary. Some aspects are controlled directly by primitive attributes, some are controlled indirectly through a bundle table. Primitives inside segments have an aspect controlled through the segment containing them, for example highlighting; primitives outside segments do not.

3.5 attribute: Attributes control the properties of output primitives and segments. There ae four types of primitive attributes: geometric, non-geometric, viewing and identification. The geometric and non-geometric attributes control the values of aspects of primitives.

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

3.7 baseline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being a lower limit of the character shape. A descender passes below this line. All baselines in a font are in the same position in the character bodies.

3.8 bundle index: An attribute of an output primitive which is an index into a bundle table. It defines workstation dependent aspects of a primitive.

3.9 bundle table: A workstation dependent table specifying aspects of one or more output primitives. GKS-3D has polyline, polymarker, text, fill area, and edge bundle tables.

3.10 capline: A horizontal line within a character body (see figure 3) which, for many character definitions, has the appearance of being the upper limit of the character shape. An ascender may pass above this line and in some languages an additional mark (for example an accent) over the character may be defined above this line. All caplines in a font are in the same position in the character bodies.

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

3.12 centre of projection: See Projection Reference Point (PRP).

3.13 centreline: A vertical line bisecting the character body (see figure 3).

3.14 character base vector: An aspect of text which defines the principal base direction of the text string. It is a two-dimensional vector in the text plane specified in the text output

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

3.16 character expansion factor: An aspect of text which specifies the deviation of character width from the defined nominal value.

3.17 character height: An aspect of text which specifies the nominal height of an upper case character.

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

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Definitions

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

3.20 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 output function.

3.21 character width: An aspect of text which specifies the nominal width of an upper case character.

3.22 choice device: A GKS-3D logical input device providing a positive integer defining one of a set of alternatives.

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

3.24 CIE L*u*v* colour model: A colour model witn an approximately uniform colour space for small colour differences and an associated colour difference formula. Its components are L* (Lightness) and u*, v* (Chromaticness). The coordinates of the associated chromaticity diagram are u* and v*.

3.25 CIE Tristimulus colour colour model: A colour model allowing the description of all colours within the colour gamut by non-negative tristimulus colour values X, Y, Z.

3.26 clipping: Removing those parts of output primitives which lie outside a specified volume.

3.27 clipping volume: Either the normalization viewport clipping volume, the view clipping volume or the intersection of the two. The normalization viewport clipping volume is a rectangular parallelepiped in normalized device coordinates (NDC). The view clipping volume is in the viewing reference coordinate system and is defined by the window, front and back planes, and projectors of the corners of the window. Output primitives within these volumes will not be clipped. Parts of output primitives on the planes forming the edges of these volumes are considered to be within the volume.

3.28 colour model: The method by which the application program describes a colour.

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

3.30 coordinate graphics; line graphics: Computer graphics in which display images are generated from display commands and coordinate data.

3.31 device coordinates (DC): A device dependent coordinate system. In GKS-3D, DC units are metres on a device capable of producing a precisely scaled image, and appropriate workstation dependent units otherwise.

3.32 device driver: The device dependent part of a GKS-3D implementation that supports a physical graphics device. The device driver generates device dependent output and handles device dependent interaction.

3.33 device space: The space defined by the addressable points of a display device.

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

3.35 display image; picture: A collection of output primitives or segments that are represented together at any one time on a display space.

3.36 display space: (1) That portion of the device space corresponding to the area available for displaying images. (2) The working space of an input device such as a digitiser.

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

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

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

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

3.40 edge: The set of boundaries of the polygons defined in the fill area set primitive.

3.41 edge bundle table: A table associating specific values for workstation dependent aspects of a fill area set primitive with an edge bundle index. In GKS-3D, this table contains entries consisting of edge flag, edgetype, edgewidth scale factor, and colour index.

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

3.43 edge type: An aspect of fill area set which indicates the style of the edges.

3.44 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.

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

3.46 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.

3.47 event queue: A time-ordered collection of event reports.

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

3.49 feedback: Output indicating to the operator the application program's interpretation of a logical input value.

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

3.51 fill area bundle table: A table associating specific values for workstation dependent aspects of fill area and fill area set primitives with bundle table index values. In GKS-3D, this table contains entries for interior style, style index, and colour index. The workstation dependent aspects of the fill area set edge are contained in the edge bundle table.

3.52 fill area set: An output primitive consisting of a set of fill areas with edge control.

3.53 front plane A plane parallel to the view plane which is specified as an N coordinate value in the view reference coordinate system. Primitives in front of the front plane lie outside the view volume.

3.54 Generalized Drawing Primitive (GDP): An output primitive which accesses implementation dependent and workstation dependent geometrical capabilities such as curve drawing.

3.55 GKS-3D level: Two values in the range 0 to 2 and a to c which together define a set of functional capabilities of GKS-3D. An implementation of GKS-3D provides precisely the functions of one level.

3.56 GKS-3D Metafile (GKSM): A sequential file that can be written or read by GKS-3D and is used for long-term storage (and for transmittal and transferral) of graphical information.

3.57 halfline: A horizontal line between the capline and the baseline within the character body (see figure 3), about which a horizontal string of characters in a font would appear centrally placed in the vertical direction. All halflines in a font are in the same position in the character bodies.

3.58 hatch interior style: One possible representation of the interior of a fill area set or fill area primitive. The interior is filled with a pattern of parallel or crossing hatch lines, selected from the workstation hatch table.

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

3.60 hidden line/hidden surface removal: Removal of those parts of output primitives which are obscured by other output primitives.

3.61 highlighting: A device independent way of emphasizing a segment by modifying its visual attributes.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Definitions

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

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

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

3.65 image: The appearance of objects after rendering.

3.66 implementation mandatory: Implementation mandatory describes a property that is required to be realized identically on all workstations of all implementations of GKS-3D.

3.67 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.

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

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

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

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

3.72 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.

3.73 locator device: A logical input device providing a position in world coordinates, a normalization transformation number, and a view index.

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

3.75 logical input value: A value delivered by a logical input device.

3.76 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.

3.77 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.

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

3.79 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.

3.80 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.

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

3.82 MI: An abbreviation for GKS-3D metafile input, a category of workstation.

3.83 MO: An abbreviation for GKS-3D metafile output, a category of workstation.

3.84 normalization transformation; window-to-viewport transformation: A transformation that maps the boundary and interior of a window to the boundary and interior of a viewport. In GKS-3D this transformation maps positions in world coordinates to normalized device coordinates.

3.85 normalized device coordinates (NDC): A coordinate specified in a device independent intermediate coordinate system, normalized to some range, typically 0 to 1. In GKS-3D, during an intermediate state the coordinates may lie outside the defined range, but associated clipping information ensures that the output does not exceed the coordinate range $[0,1]\times[0,1]\times[0,1]$.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

3.86 normalized projection coordinates (NPC): The device-dependent three dimensional Cartesian coordinate system in which the composition of images is specified to the graphics system. The projection viewport and workstation window are specified in NPC space.

3.87 oblique view: A parallel projection whose projectors are not perpendicular to the view plane or a perspective projection where the projector through the centre of the view window is not perpendicular to the view plane.

3.88 operator: Person manipulating physical input devices so as to change the measures of logical input devices and cause their triggers to fire.

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

3.90 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.

3.91 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.

3.92 pick device: A GKS-3D logical input device providing the pick identifier attached to an output primitive and the associated segment name.

3.93 pick identifier: A name, attached to individual output primitives within a segment, and returned by the pick device. The same pick identifier can be assigned to different output primitives.

3.94 pixel; picture element: The smallest element of a display space that can be independently assigned a colour or intensity.

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

3.96 polyline bundle table: A table associating specific values for all workstation dependent aspects of a polyline primitive with a polyline bundle index. In GKS-3D, this table contains entries consisting of linetype, linewidth scale factor, and colour index.

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

3.98 polymarker bundle table: A table associating specific values for all workstation dependent aspects of a polymarker primitive with a polymarker bundle index. In GKS-3D, this table contains entries consisting of marker type, marker size scale factor, and colour index.

3.99 primitive attribute: Primitive attribute values (for output primitives) are selected by the application in a workstation independent manner, but can have workstation dependent effects.

3.100 projection plane: See view plane.

3.101 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.

3.102 projector: A conceptual line which passes through each point of an object and intersects the view plane.

3.103 prompt: Output to the operator indicating that a specific logical input device is available.

3.104 raster graphics: Computer graphics in which a display image is composed of an array of pixels arranged in rows and columns.

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

3.106 rotation: Turning all or part of a display image about an axis. In GKS-3D, this capability is available via the segment and viewing transformations.

3.107 scaling; zooming: Enlarging or reducing all or part of a display image by multiplying the coordinates of display elements by a constant value. In GKS-3D, this capability is available via the

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Definitions

segment and viewing transformations.

NOTE - For different scaling in three orthogonal directions three constant values are required.

3.108 segment: A collection of display elements that can be manipulated as a unit.

3.109 segment attributes: Attributes that apply only to segments. In GKS-3D, segment attributes are visibility, highlighting, detectability, segment priority, and segment transformation.

3.110 segment priority: A segment attribute used to determine which of several overlapping segments takes precedence for graphic output and input.

3.111 segment transformation: A transformation that causes the display elements defined by a segment to appear with varying position (translation), size (scale), and/or orientation (rotation) on the display space.

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

3.113 string device: A logical input device providing a string.

3.114 stroke device: A logical input device providing a sequence of points in world coordinates and an associated normalization transformation number and view index.

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

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

3.117 text bundle table: A table associating specific values for all workstation dependent aspects of a text primitive with a text bundle index. In GKS-3D, this table contains entries consisting of text font and precision, character expansion factor, character spacing and colour index.

3.118 text direction vectors: Two vectors, specified as part of the text output function, which, together with the text position, define the text plane.

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

3.120 text font and precision: An aspect of text in GKS-3D, having two components, font and precision, which together determine the shape of the characters being output, on a particular workstation. In addition, the precision describes the fidelity with which the other text aspects match those requested by an application program. In order of increasing fidelity, the precisions are: STRING, CHAR and STROKE.

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

3.122 text plane: The plane in which a text primitive appears.

3.123 text position: A point, specified as part of the text output function, which, together with the text direction vectors, determine the text plane. It also determines, together with the text alignment, the location of the string.

3.124 translation; shift: The application of a constant displacement to the position of all or part of a display image. In GKS-3D, this capability is restricted to segments.

3.125 trigger: A physical input device or set of devices that an operator can use to indicate significant moments in time.

3.126 valuator device: A logical input device providing a real number.

3.127 view index: The attribute of all primitives that selects the viewing representation applied to the primitives at display time.

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

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

3.129 view orientation matrix: A 4×4 matrix which specifies the transformation of points from world coordinates to view reference coordinates.

3.130 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.

3.131 view plane distance: The distance of the view plane from the view reference point measured along the view plane normal vector in view reference system coordinates.

3.132 view plane normal: A vector in NDC or world coordinates, relative to the view reference point, which defines the N coordinate axis in view reference coordinate space.

3.133 view reference coordinate system (VRC): 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. The unit vectors of this system are of equal length to that of the NDC system.

3.134 view reference point: A point in NDC or world coordinates which defines the view reference coordinate system origin.

3.135 view table: The workstation dependent table of views found on each workstation. It is referenced by the workstation independent view index attribute.

3.136 view up vector: A vector defined in NDC or 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 NDC or world coordinate space will appear vertical in the final image.

3.137 view volume: The view volume is detemined 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 parallepiped. In two dimensional applications, the view volume reduces to the view window.

3.138 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.

3.139 viewing transformation: A transformation that maps positions in NDC coordinates to positions in NPC coordinates.

3.140 viewport: An application program specified part of normalized device coordinate space. In GKS-3D, this definition is restricted to a rectangular parallelepiped region of normalized device coordinate space used in the definition of the normalization transformation.

3.141 window: A predefined part of a virtual space. In GKS-3D, this definition is restricted to a rectangular parallelepiped region of the world coordinate space used for the definition of the normalization transformation.

3.142 workstation: GKS-3D is based on the concept of abstract graphical workstations, which provide the logical interface through which the application program controls physical devices.

3.143 workstation dependent segment storage (WDSS): Segment storage on a workstation that is used for graphical output. Segments cannot be transferred from WDSS to another workstation.

3.144 workstation independent segment storage (WISS): A special workstation type, where segments can be stored and later transferred to other workstations.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

Definitions

3.145 workstation mandatory: Workstation mandatory describes a property that is required to be realized identically on all workstations of a GKS-3D implementation.

3.146 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.

3.147 workstation viewport: A rectangular parallelepiped in device coordinate space. All graphics output appears within this volume.

3.148 workstation window: A rectangular parallelepiped in normalized projection coordinate space. Its contents are mapped by the workstation transformation into the workstation viewport.

3.149 world coordinate (WC): A device independent Cartesian coordinate system used by the application program for specifying graphical input and output.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

4 The Graphical Kernel System for Three Dimensions

4.1 About this International Standard

4.1.1 Specification

The set of functions known as the Graphical Kernel System for Three Dimensions shall be as described in clauses 4, 5 and 6 and annexes A and B. These functions are organized in nine upward compatible levels with increasing capabilities as described in 4.10. An implementation of GKS-3D shall implement precisely the functions of one level. A GKS-3D implementation shall be invalid if it lies between two defined levels or outside the defined levels. In an implementation, all graphical capabilities that can be addressed by GKS-3D functions shall be used only via GKS-3D.

4.1.2 Registration[1)]

For certain parameters of the functions, GKS-3D defines value ranges as being reserved for registration. The meanings of these values will be defined using the established procedures. 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 directives, the ISO Council has designated the National Bureau of Standards (Institute of Computer Sciences and Technology) A-266, Technology Building, Gaithersburg, MD 20899, USA to act as the registration authority.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

4.2 Introduction to GKS-3D

The Graphical Kernel System for Three Dimensions (GKS-3D) provides a functional interface between an application program and a configuration of graphical input and output devices. The functional interface contains all basic functions for interactive and non-interactive 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, FILL AREA, FILL AREA SET, CELL ARRAY, GENERALIZED DRAWING PRIMITIVE), and uniform input classes (LOCATOR, STROKE, VALUATOR, CHOICE, PICK, STRING) is obtained.

In 4.3 the concepts of basic output, input and the organization of input and output sequences are outlined. A central concept both for structuring GKS-3D and for realizing device independence is introduced, called the workstation.

The facilities for picture manipulation and change are introduced via the segment facilities, the dynamic attributes and the transformations. The integral control over all these methods for change is further explained in 4.5.3 on workstations.

The concept of multiple workstations allows simultaneous output to and input from various display systems. Facilities for internal and external storage are provided by special workstations together with the possibility of transferring graphical entities directly from the special workstation for internal storage to other workstations.

Not every GKS-3D implementation needs to support the full set of functions. Nine levels are defined to meet the different requirements of graphics systems. Each GKS-3D implementation precisely provides the functions of one level. The levels are upward compatible.

GKS-3D defines only a language independent nucleus of a graphics system. For integration into a language, GKS-3D is embedded in a language dependent layer containing the language conventions, for example, parameter and name assignment.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

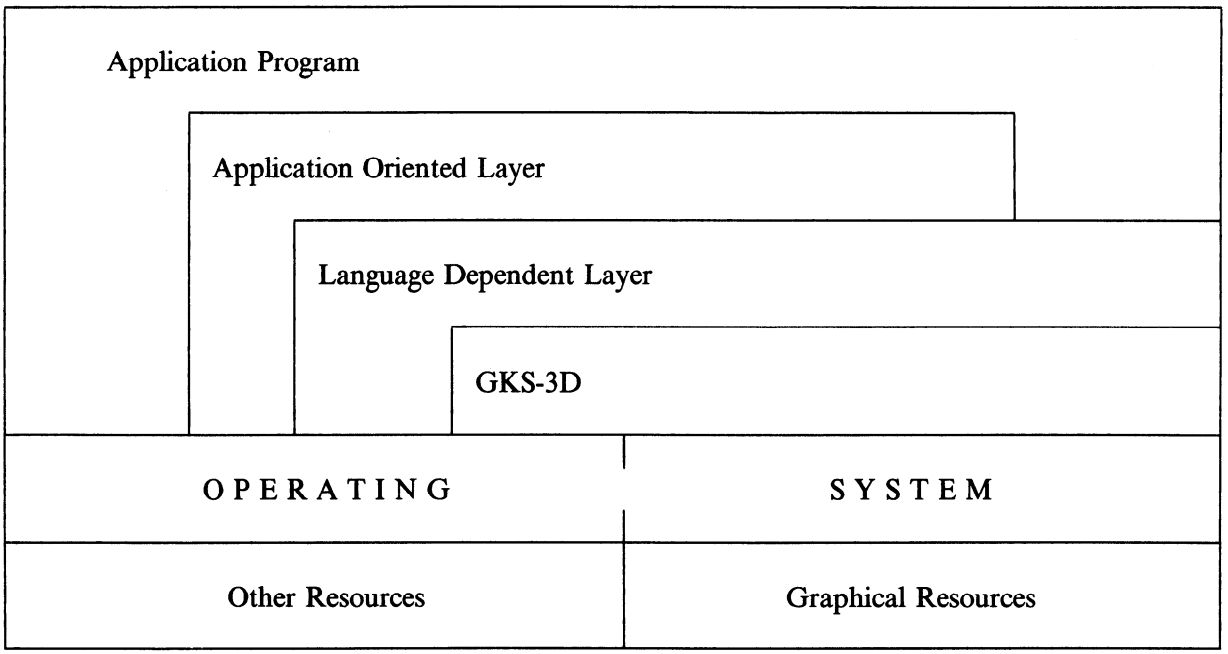


Figure 1 - Layer model of GKS-3D

The layer model represented in figure 1 illustrates the role of GKS-3D 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. All workstation capabilities that can be addressed by GKS-3D functions are used only via GKS-3D.

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

4.3 Concepts

The graphical output that is generated by GKS-3D 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 printing character strings. The attributes control the aspects of the output primitives on a device, such as linestyle, colour, character height and pick identifier. Non-geometric aspects, such as colour but not character height, can be controlled for each workstation individually, 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 GKS-3D 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 space, such as prompts and echoes, is controlled by GKS-3D for each logical input device individually.

The two abstract concepts (abstract output and abstract input) are the building blocks of a so-called abstract workstation. A workstation of GKS-3D represents a unit consisting of zero or one display spaces 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. Conceptually, all workstations are three dimensional in GKS-3D.

The geometrical information (coordinates) contained in output primitives, attributes and logical input values (locators and strokes) can be subjected to transformations. These transformations perform mappings between five coordinate systems, namely:

a) World Coordinates (WC), used by the application programmer;

b) Normalized Device Coordinates (NDC), used to define a uniform coordinate system for all workstations;

c) View Reference Coordinates (VRC), used to define the parameters of a view;

d) Normalized Projection Coordinates (NPC), used to facilitate hidden line/hidden surface computations and composition of different views;

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

Output primitives and attributes are mapped from WC to NDC by normalization transformations, from NDC to NDC by segment transformations (see next paragraph), from NDC to VRC by view orientation transformations, from VRC to NPC by view mapping transformations, and from NPC to DC by workstation transformations. Locator and stroke input is mapped by the inverse of these transformations from DC to WC.

Output primitives and primitive attributes may be grouped together in a segment. Segments are the units for manipulation and change. Manipulation includes creation, deletion, and renaming. Change includes transforming a segment, making a segment visible or invisible, and highlighting a segment. Segments also form the basis for workstation independent storage of pictures at run time. Via this storage, which is set up as a special workstation called workstation independent segment storage, segments can be inserted and transferred to other workstations.

The attributes which control the appearance of parts of the picture (output primitives, segments, prompt and echo types of input devices) 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. The primitive attributes include all geometric

AVA Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

aspects of primitives, such as character height for text and pattern size for fill area. In addition, the non-geometric aspects of primitives are controlled by the primitive attributes in one of two ways. A non-geometric aspect of a primitive can be specified either by an attribute that specifies one or more aspects of the primitive by an index that points to a workstation dependent representation (set of values) or by an attribute that specifies each aspect independent of the workstation. The former is referred to as bundled specification and the latter is referred to as individual specification. In addition, an index is bound to the primitive which selects the view and projection transformations in a workstation dependent manner.

Hidden lines or hidden surfaces may be removed using a workstation dependent method (see 4.5.7).

Workstation attributes include the actual representations on a workstation pointed to by indices used in bundled specification of non-geometric aspects. 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 the control over deferral of picture change. Workstation attributes can be reset dynamically.

The appearance of segments is controlled by segment attributes, which are segment transformation, visibility, highlighting, segment priority, and detectability. These may be reset dynamically. Segment attributes can be a basis for feedback during manipulations (for example, highlighting).

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 a volume within the display space 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: SAMPLE input is acquired directly by the application program; REQUEST input is produced by the operator in direct response to the application program; EVENT input is generated asynchronously by the operator and is collected in a queue for the application program.

At run time GKS-3D can be in one of five different operating states. Associated with each state are the set of GKS-3D functions allowed in this state, and a set of state variables. The operating state concept and the state variables allow for proper specification of initializations (for example, at OPEN WORKSTATION) and the effect of various functions, especially with respect to the maintenance of device independence. One special set of functions called inquiry functions is allowed in all states. They give read-only access to the state lists. In this way useful information can be provided when errors occur. Other inquiry functions allow read-only access to the workstation descriptions, to allow the application program to adapt to particular workstation capabilities. Inquiry functions never cause errors. Instead they return information specifying whether a valid inquiry was made.

GKS-3D provides an interface to a system for filing graphical information for the purpose of external long term storage and exchange. The interface consists of a GKS-3D Metafile output workstation, which writes to a so-called graphics metafile (which is sequential), and a GKS-3D Metafile input workstation, which reads from the metafile. In addition to the normal functions for output to workstations, a GKS-3D Metafile output workstation may accept items containing non-graphical information. Input from a metafile is controlled by read and interpret functions which have the same effect as invoking the corresponding functions directly from the application program.

GKS-3D provides graphic functions for primitives defined in a three dimensional space whereas ISO 7942 provides functions for primitives defined in a plane. Thus, ISO 7942 functions are a special

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

case of the corresponding functions in GKS-3D. The functions common to ISO 7942 and GKS-3D are called the 2D functions.

GKS-3D has a set of output primitives (see 4.4.1) which can be generated either by a 3D function or by a 2D function (e.g., a polyline primitive is generated by the 3D function POLYLINE 3 or the 2D function POLYLINE). However, 2D functions used in a 3D environment can be affected by 3D functions, for example, primitives generated using 2D functions will be transformed by 3D transformation matrices.

The primitives generated by the 2D functions are considered to lie in the Z=0 plane. The default view is an orthographic projection onto the Z=0 plane. The implication is that application programs run on a GKS-3D implementation produce the same results as on an ISO 7942 implementation.

The mapping between the 2D functions and the corresponding 3D functions is described in 4.14.

AVA Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

4.4 Graphical output

4.4.1 Output primitives

The graphical information that is generated by GKS-3D and routed to all active workstations is built up of basic pieces called output primitives. GKS-3D provides seven output primitives:

a) POLYLINE: GKS-3D generates a set of connected lines defined by a point sequence.

b) POLYMARKER: GKS-3D generates symbols of one type centred at given positions.

c) TEXT: GKS-3D generates a character string at a given position.

d) FILL AREA: GKS-3D generates a polygonal area which may be hollow or filled with a uniform colour, a pattern, or a hatch style.

e) FILL AREA SET: GKS-3D generates a set of polygonal areas which may be hollow or filled with a uniform colour, a pattern, or a hatch style. This allows for specifying areas with holes or disjoint regions that must be treated as a single entity, particularly for hidden line/hidden surface removal computation.

f) CELL ARRAY: GKS-3D generates an array of pixels with individual colours.

g) GENERALIZED DRAWING PRIMITIVE(GDP): GKS-3D addresses special geometrical 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. GKS-3D 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 2 sides and zero thickness. The viewing transformation determines which side of the primitive is seen.

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. Similarly, the plane in which the cell array primitive lies is defined by three points contained in the definition of the cell array primitive. 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 are coplanar is the responsibility of the application program. Implementations are not required to check coplanarity. While slightly non-coplanar points (e.g., owing to numerical roundoff errors) should be mapped to a plane, the effect of large deviations is implementation dependent.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

4.4.2 Output primitive attributes

Each output primitive potentially has four types of attribute (geometric, non-geometric, viewing and identification). The first three attribute types determine the exact appearance of the output primitive while the fourth attribute type is used in connection with input. The values of these attributes are set modally and are recorded in the GKS-3D state list. 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 (that is, when one of the GKS-3D output primitive functions is invoked) 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 which affect the shape or size of the whole primitive (for example, CHARACTER HEIGHT for TEXT). Hence, they are sometimes referred to as geometric attributes. Attributes of this type are workstation independent and, if they represent coordinate data (points or displacements), are expressed in world coordinates (for example, CHARACTER HEIGHT is expressed in world coordinates but TEXT PATH takes one of a set of enumerated values). They are defined separately for each primitive and a primitive may have zero, one or many geometric attributes.

Current values of (workstation independent) geometric attributes, which are expressed in world coordinates, are stored in world coordinates. When they 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. Hence, current values are unaffected by changes in the normalization, viewing and workstation transformation.

Attributes of the second type control the non-geometric aspects of primitives; these are aspects which merely 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 in one of two ways, namely via a bundle or individually.

Bundled aspects are controlled 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 this method of specification and each 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 in the GKS-3D 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. Since the bundles are set separately for each workstation, the values of their components are restricted to the valid values for that workstation. In the case of individual attribute specification, such restrictions are not imposed. Default actions for the display of a primitive are defined to occur if it is created with a value of an individually specified attribute that is invalid on a particular workstation.

As indicated 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

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

function. For example, if one GDP function is essentially 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 and so does not use a bundle table nor does it have a set of individually specified attributes.

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 in the GKS-3D state list. There is one ASF for each non-geometric aspect. The initial values of all 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 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 the ASFs are INDIVIDUAL.

The flags may be set at any time when GKS-3D is open by the functions SET ASPECT SOURCE FLAGS and SET ASPECT SOURCE FLAGS 3. This enables some non-geometric aspects of a primitive to be specified individually and others via a bundle.

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 that primitive.

d) If the ASF for an aspect is BUNDLED, the value used on a workstation is obtained via the bundle table for that primitive on the workstation; the corresponding component of the bundle, pointed to by the bundle index, is used.

If colour is a non-geometric aspect of a primitive, then it is specified as an index into a separate colour table. There is only one colour table per 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 fixed list (for example, linetypes for polyline).

There are two attributes of the third type, namely VIEW INDEX and HLHSR IDENTIFIER. The attribute VIEW INDEX is a pointer to a view table entry at a workstation. There is one view table per workstation.

The attribute HLHSR IDENTIFIER supplies HLHSR information to the workstation. The interpretation of the identifier is implementation dependent (see 4.5.7). 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 is precisely one attribute of the fourth type per primitive, namely PICK IDENTIFIER. This is used for identifying a primitive, or a group of primitives, in a segment when that segment is picked.

The attributes which apply to each output primitive (attributes controlling non-geometric aspects, geometric attributes, view attributes and PICK IDENTIFIER) are:

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

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

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

g) TEXT:
TEXT INDEX
TEXT FONT AND PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR INDEX
TEXT FONT AND PRECISION ASF
CHARACTER EXPANSION FACTOR ASF
CHARACTER SPACING ASF
TEXT COLOUR INDEX ASF
CHARACTER HEIGHT
CHARACTER UP VECTOR
TEXT PATH
TEXT ALIGNMENT
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

h) FILL AREA:
FILL AREA INDEX
FILL AREA INTERIOR STYLE
FILL AREA STYLE INDEX
FILL AREA COLOUR INDEX
FILL AREA INTERIOR STYLE ASF
FILL AREA STYLE INDEX ASF
FILL AREA COLOUR INDEX ASF
PATTERN SIZE
PATTERN REFERENCE POINT AND VECTORS
VIEW INDEX
HLHSR IDENTIFIER
PICK IDENTIFIER

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

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

Figure 2 shows the binding of the attributes.

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

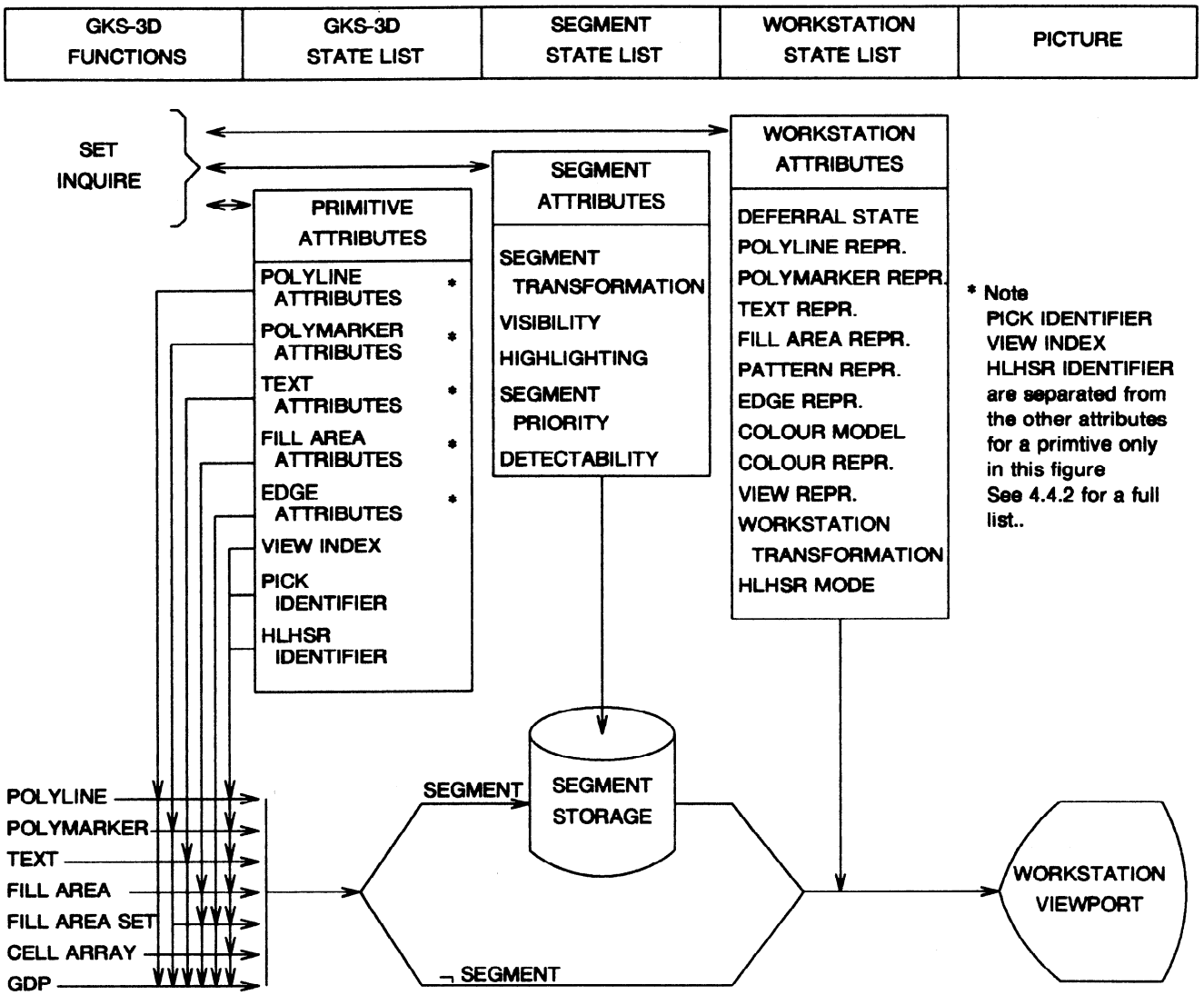


Figure 2 - Binding of attributes

Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

The attributes for each primitive, other than PICK IDENTIFIER, are described in 4.4.3 to 4.4.13. PICK IDENTIFIER is described in more detail in 4.7.1. VIEW INDEX and HLHSR IDENTIFIER are described in more detail in 4.4.12 and 4.4.13. 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 which appear in upper case (for example, the aspect linetype but the individually specified attribute LINETYPE).

The entries in the bundle, pattern, colour, and view tables may be set separately for each workstation. Some standard 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 to be predefined for each output type workstation. At least those predefined entries with indices up to the minimum number of predefined entries at a given level (see 4.10.3) are distinguishable from each other. Other combinations of values can be specified by the SET <bundle | PATTERN | COLOUR> REPRESENTATION and SET VIEW REPRESENTATION 3 functions, possibly after inquiring the workstation capabilities. The tables, which are on every workstation of category OUTPUT, OUTIN or MO (i.e. they are workstation attributes), are:

polyline bundle table
polymarker bundle table
text bundle table
fill area bundle table
edge bundle table
pattern table
colour table
view table

The values in these tables may be (dynamically) changed. In fact, the only way of changing the aspects of a primitive which are stored in a bundle table is by changing that table. However, note that a change in a bundle table entry can only be reflected in a displayed primitive if the values of the corresponding ASFs (of that primitive) for the aspects in the bundle table are BUNDLED. The entry 'dynamic modification accepted' in the workstation description table indicates which changes

l) lead to an implicit regeneration (may be deferred) (IRG);

m) can be performed immediately (IMM).

The deferral state is explained in more detail in 4.5.3. If changes can be performed immediately, those changes may affect primitives outside segments in addition to those inside segments.

4.4.3 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 solid, dashed, dotted and dashed-dotted. Every workstation of category

AVA Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

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 0 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 a 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.

4.4.4 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 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 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 0 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. Marker type 1 is always displayed as the smallest displayable dot.

The marker is visible if, and only if, the marker position is within the clipping volume. The clipping of partially visible markers is workstation dependent. Since the polymarker has no geometric attributes, transformations affect only the marker position and not the appearance of the marker glyph.

4.4.5 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. This coordinate system, which is known as the text local coordinate (TLC) system, will be defined in the next paragraph.

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 local coordinate system is at the point P. The positive X axis of the 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. By this definition, any point in the text plane can be represented by two reals r_1 and r_2, since it is uniquely defined by the expression $P + r_1E_1 + r_2E_2$, where E_1 and E_2 are two unit vectors parallel to the positive X axis and Y axis, respectively. This means that there is a 1-1 correspondence between a pair of two reals and a point in the text plane. In a similar way, there is a 1-1 correspondence between a pair of two reals and a vector parallel to the text plane. This correspondence holds especially for the 2D vectors

AVA Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

CHARACTER UP VECTOR and the CHARACTER BASE VECTOR, which will be later be defined in this subclause.

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 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 representation of text at the workstation is controlled by the TEXT INDEX, or the set of individually specified text attributes (TEXT FONT AND PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, and TEXT COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for text font and precision, character expansion factor, character spacing and text colour index. The TEXT INDEX is a pointer into the text bundle table, each entry of which contains values for text font and precision, character expansion factor, character spacing and text colour index.

Precise control of the appearance of TEXT on a workstation is provided by the following aspects: CHARACTER HEIGHT, CHARACTER WIDTH, character expansion factor, TEXT PATH, CHARACTER UP VECTOR, CHARACTER BASE VECTOR, character spacing and TEXT ALIGNMENT. However, the use of these values in displaying text is determined by the setting of the text font and precision aspect (font and precision are two components of the same aspect). The CHARACTER HEIGHT specifies the nominal height of a capital letter character. The CHARACTER WIDTH specifies the nominal width of a character; the actual width depends upon the width to height ratio of the character indicated by the font designer and may vary from character to character. The character expansion factor specifies the deviation of the width to height ratio of the character from the ratio indicated by the font designer. The CHARACTER UP VECTOR gives the up direction of a character. The CHARACTER BASE VECTOR gives the direction of the baseline of a character. Only the directions, not the lengths, of these vectors are relevant. TEXT PATH has the possible values RIGHT, LEFT, UP and DOWN. It specifies the writing direction of the text string. For RIGHT, the text string is written along a baseline in the direction of the CHARACTER BASE VECTOR. For LEFT, the baseline direction is the opposite direction of the CHARACTER BASE VECTOR. For UP, the character path coincides with the direction of the CHARACTER UP VECTOR. For DOWN, it is the opposite direction of the CHARACTER UP VECTOR. For the UP and DOWN text path directions the characters are arranged so that the centres of the character bodies are on a straight line in the direction of the CHARACTER UP VECTOR.

The character spacing value specifies how much additional space is to be inserted between two adjacent character bodies. If the value of character spacing is zero, the character bodies are arranged one after the other along the TEXT PATH, without any additional space between. A positive value of character spacing will insert additional space between character bodies. A negative value of character spacing will cause adjacent character bodies to overlap. Character spacing is specified as a fraction of the font nominal character height.

The effect of the aspects CHARACTER HEIGHT, CHARACTER WIDTH, character expansion factor, TEXT PATH, character spacing and text font is to define an (imaginary) rectangle with its

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

sides parallel to the x and y axes, enclosing the text. The bounds of this enclosing rectangle are as follows. For TEXT PATH = LEFT or RIGHT, the height of the rectangle is the height of the character body of the specified font; the left side of the rectangle is the left side of the character body of the leftmost character and the right side of the rectangle is the right side of the character body of the rightmost character. For TEXT PATH = UP or DOWN, the top of the rectangle is the top of the character body of the topmost character and similarly, the bottom of the rectangle is the bottom of the bottommost character; the width of the rectangle is the width of the widest character in the specified font.

The effect of the CHARACTER UP VECTOR and CHARACTER BASE VECTOR attributes is to transform the enclosing rectangle, thus defining an enclosing parallelogram, the text extent parallelogram (the rectangle has been rotated and sheared).

The TEXT ALIGNMENT attribute controls the positioning of this text extent parallelogram in relation to the text position. For simplicity the TEXT ALIGNMENT is described in terms of the default CHARACTER UP VECTOR and CHARACTER BASE VECTOR, when the text extent parallelogram is actually a rectangle. The horizontal component of TEXT ALIGNMENT has four values: LEFT, CENTRE, RIGHT and NORMAL. If the horizontal component is LEFT, the left side of the text extent parallelogram passes through the text position. Similarly, if the value is RIGHT, the right side of the text extent parallelogram passes through the text position. If the horizontal component is CENTRE, the text position lies midway between the left and right sides of the text extent parallelogram. Thus, if TEXT PATH = UP or DOWN, the straight line passing through the centrelines of the characters also passes through the text position. The vertical component of TEXT ALIGNMENT has six values: TOP, CAP, HALF, BASE, BOTTOM and NORMAL. These each correspond to one of the font specific horizontal lines in the definition of a character (see figure 3). A value of TOP causes the top of the text extent parallelogram to pass through the text position. A value of CAP causes the text position to lie on the capline of the whole string (TEXT PATH = LEFT or RIGHT) or on the capline of the topmost character in the string (TEXT PATH = UP or DOWN). A value of HALF causes the text position to lie on the halfline of the whole string (TEXT PATH = LEFT or RIGHT) or on a line halfway between the halflines of the top and bottom characters (TEXT PATH = UP or DOWN). A value of BASE causes the text position to lie on the baseline of the whole string (TEXT PATH = LEFT or RIGHT) or on the baseline of the bottom character in the string (TEXT PATH = UP or DOWN). A value of BOTTOM causes the bottom of the text extent parallelogram to pass through the text position.

In the general case, the orientation referred to as horizontal is that of the CHARACTER BASE VECTOR with RIGHT representing direction of that vector and LEFT being opposite to it. Similarly the orientation referred to as vertical is that of the CHARACTER UP VECTOR with UP representing the direction of that vector and DOWN being opposite to it.

Either component of TEXT ALIGNMENT can take the value NORMAL. For each value of TEXT PATH, the effect of a particular component being NORMAL is equivalent to one of the other values of that component. In each case, the equivalent alignment value is chosen to achieve a natural alignment for that TEXT PATH value. The complete list of equivalent values is:

TEXT PATH	NORMAL Horizontal and Vertical Alignments
RIGHT	(LEFT, BASE)
LEFT	(RIGHT, BASE)
UP	(CENTRE, BASE)
DOWN	(CENTRE, TOP)

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

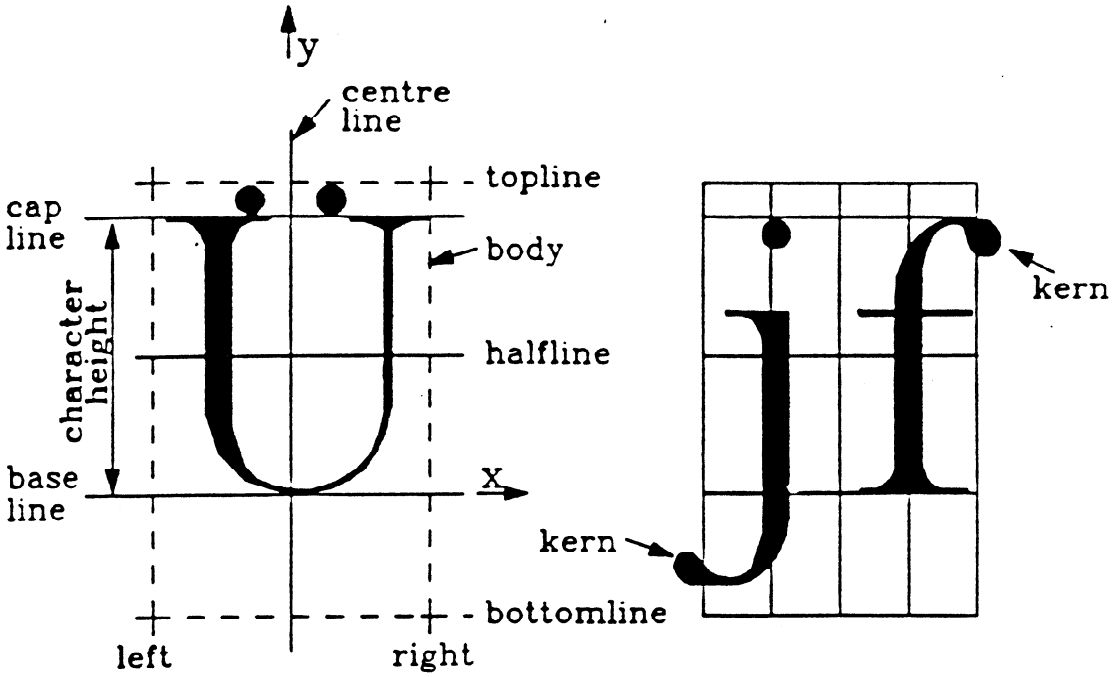


Figure 3 - Font description coordinate system

The initial values of the geometric text attributes are:

CHARACTER HEIGHT	TLC	0.01 (i.e. 1% of the height of the default window)
CHARACTER UP VECTOR	TLC	(0,1)
TEXT PATH		RIGHT
TEXT ALIGNMENT		(NORMAL, NORMAL)

and the initial values of the implicitly specified geometric text attributes are:

CHARACTER WIDTH	TLC	0.01 (i.e. the same value as the initial value of CHARACTER HEIGHT)
CHARACTER BASE VECTOR	TLC	(1,0)

Text font and precision together constitute one aspect. The text font value is used to select a particular font on the workstation. Every workstation supports at least one font that is able to generate a graphical representation of the characters defined in ISO 646. This is font number 1. Font numbers greater than 1 are reserved for registration (see 4.1.2). Font numbers less than 0 may be supported but are implementation dependent.

The text precision value is used to select the 'closeness' of the text representation at the workstation in relation to that defined by the workstation independent text attributes and the transformation and clipping currently applicable. The text precision value has the following possible values:

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

a) STRING: The TEXT character string is generated in the requested text font and is positioned by aligning the TEXT output primitive at the given text position. CHARACTER HEIGHT, CHARACTER WIDTH and character expansion factor are evaluated as closely as reasonable, given the capabilities of the workstation. CHARACTER UP VECTOR, CHARACTER BASE VECTOR, TEXT PATH, TEXT ALIGNMENT and character spacing, need not be used. Clipping is done in an implementation and workstation dependent way.

b) CHAR: The TEXT character string is generated in the requested text font. For the representation of each individual character, the aspects CHARACTER HEIGHT, CHARACTER WIDTH, the up direction of the CHARACTER UP VECTOR, the baseline direction of the CHARACTER BASE VECTOR, and character expansion factor are evaluated as closely as possible, in a workstation dependent way. The spacing used between character bodies is evaluated exactly; the character body, for this purpose, is an ideal character body, calculated precisely from the text aspects and the font dimensions. The position of the resulting text extent parallelogram is determined by the TEXT ALIGNMENT and the text position. Clipping is performed at least on a character by character basis.

c) STROKE: The text character string in the requested text font is displayed at the text position by applying all text aspects. The character string is clipped exactly at the clipping volume.

STROKE precision does not necessarily mean vector strokes; as long as the representation adheres to the rules governing STROKE precision, the font may be realized in any form, for example by raster fonts.

A GKS-3D output level 0 implementation supports text precisions STRING and CHAR. Above output level 0, all text precisions are supported as follows. A workstation may use a higher precision than the one requested for this purpose i.e. if STROKE precision is supported in a particular font, the implication is that both STRING and CHAR precision are available in that font. However it is not necessary for a workstation to support all precisions for a given font (i.e. for a given font, STROKE can be missing or both STROKE and CHAR can be missing). Text font and precision are workstation mandatory. That is, for any GKS-3D level supporting a STROKE precision font, every workstation of a particular installation supports at least one STROKE precision text font. This is font number 1, containing the character set defined by ISO 646. In addition, font number 1 shall be monospaced with a workstation dependent aspect ratio. This implies that, for STROKE precision text, some sort of software character generator is required for those implementations that have inadequate hardware. Not all workstations need to support all fonts, but for those that do, the same font number is used to select that font on all workstations of a particular installation.

Fonts are defined only within the GKS-3D implementation. The font designer specifies the shape of the symbol representing each character in a local 2D cartesian font coordinate system. Fonts are either monospaced or proportionally spaced. Each character in a font coordinate system has an associated character body, a font baseline, a font halfline, a capline and a centreline (see figure 3). For monospaced fonts, the character bodies of all characters have the same size. For proportionally spaced fonts, the width of the bodies may differ from character to character. The character body edges are parallel to the axes of the font coordinate system. The font baseline, the font halfline and the capline are parallel to the x-axis of the font coordinate system, and within the vertical extent of the body. The position of the font halfline is defined by the font designer for use in aligning text strings. The centreline is parallel to the y-axis and bisects the body. Their exact positions are specified by the font designer.

AVA Documents Document Sharing Hub

Full document details: BS ISO 8805:1988

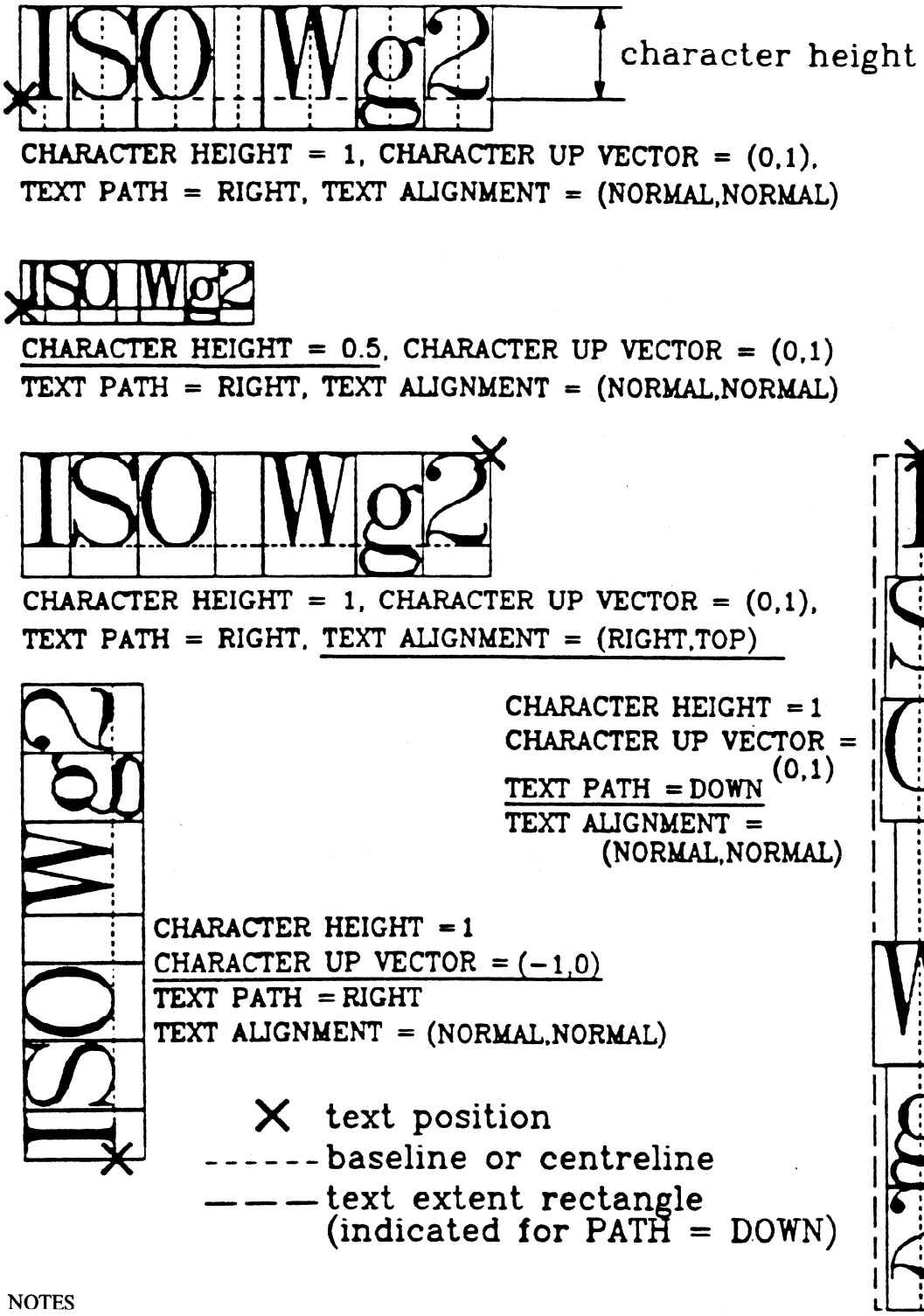
The height of a character in the font coordinate system is given by the height from the font base line to the capline. The width of a character is given by the width of the character body. The width of a character may include space on either side of the character and this space is generally evenly split between the left and right sides of the character. It is assumed that the characters lie within their body, except that kerned characters may exceed the side limits of the character body.

In general, the top limits of the bodies for a font are identical with, or very close to, the typographical capline or ascender line, and the bottom limit to the descender line. The space, if any, between the topline and the capline may be used for an additional mark over the character, for example an accent. However, these and other details are purely for the use of the font designer. The intention is only that characters placed with their bodies touching in the horizontal direction should give an appearance of good normal spacing, and characters touching in the vertical direction will avoid clashes between ascenders and descenders (typographically 'set solid').

Since the values of CHARACTER HEIGHT, CHARACTER WIDTH, CHARACTER UP VECTOR and CHARACTER BASE VECTOR are given in text local coordinate units, but the characters are generated on the workstation in device coordinates, using the workstation dependent font and precision, the geometric attributes need to be transformed in such a way that the workstation can generate the characters in the way intended.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988



NOTES

1 Examples are illustrated with STROKE precision, a character expansion factor of 1 and a zero character spacing.

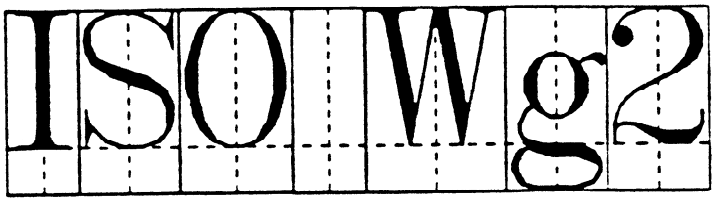
2 Capline = topline in these examples.

3 Changed attributes are underlined.

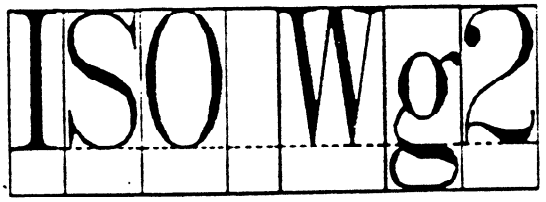
Figure 4 - Effects of changes in geometric text attributes

AVA Documents
Document Sharing Hub

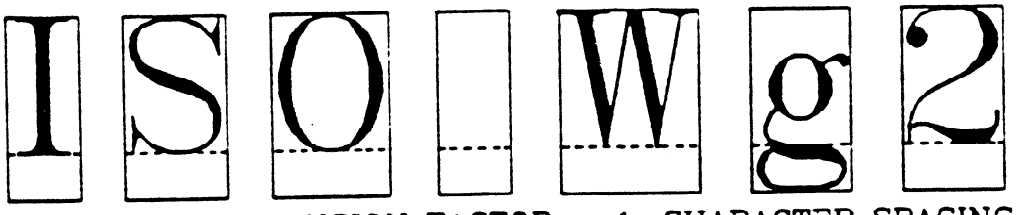
Full document details: BS ISO 8805:1988



CHARACTER EXPANSION FACTOR = 1, CHARACTER SPACING = 0



CHARACTER EXPANSION FACTOR = 0.75, CHARACTER SPACING = 0



CHARACTER EXPANSION FACTOR = 1, CHARACTER SPACING = 0.3

NOTES

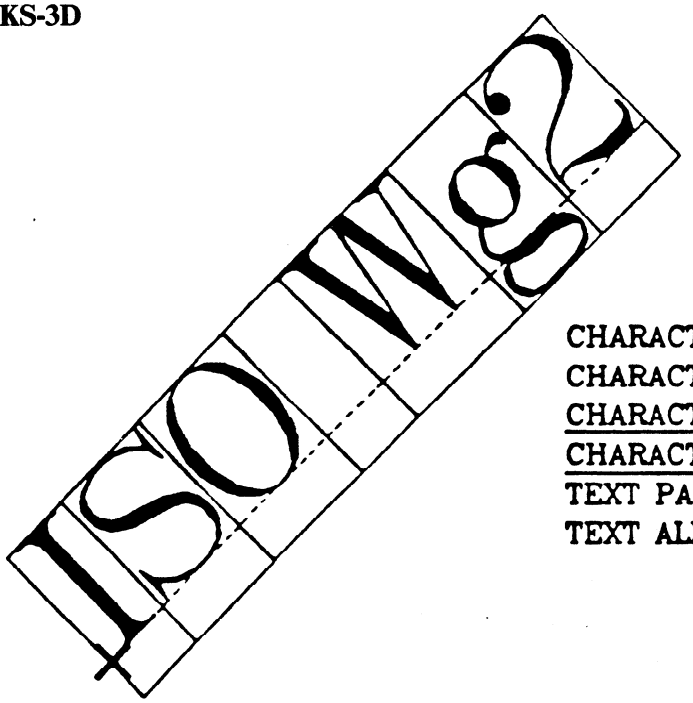
1 Examples are illustrated with default values of the geometric text attributes and with STROKE precision.

2 Changed aspects are underlined.

Figure 5 - Effects of changes in non-geometric text aspects

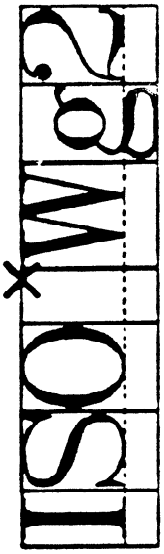
AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

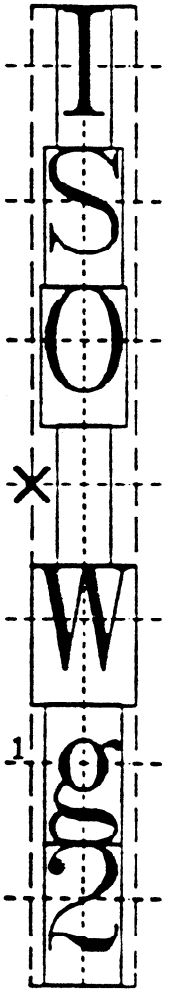


CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1.414
CHARACTER UP VECTOR = (−1,1)
TEXT PATH = RIGHT
TEXT ALIGNMENT = (NORMAL,NORMAL)

CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = −0.3
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (0,1)
TEXT PATH = LEFT
TEXT ALIGNMENT = (NORMAL,NORMAL)



CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (−1,0)
TEXT PATH = RIGHT
TEXT ALIGNMENT = (CENTRE,TOP)



CHARACTER EXPANSION FACTOR = 1
CHARACTER SPACING = 0
CHARACTER HEIGHT = 1
CHARACTER UP VECTOR = (0,1)
TEXT PATH = DOWN
TEXT ALIGNMENT = (LEFT,HALF)

NOTES

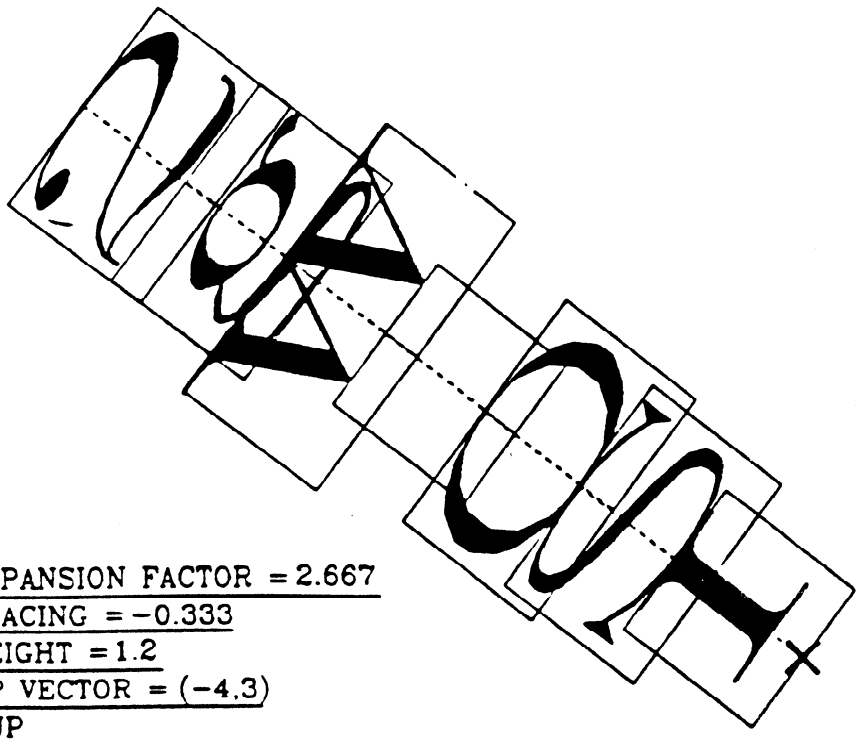
1 Changes from the top example of figures 4 and 5 are underlined.

2 In the last example, halflines of all characters are shown.

Figure 6 - Effects of combined changes in text aspects

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988



CHARACTER EXPANSION FACTOR = 2.667
CHARACTER SPACING = −0.333
CHARACTER HEIGHT = 1.2
CHARACTER UP VECTOR = (−4,3)
TEXT PATH = UP
TEXT ALIGNMENT = (CENTRE,BOTTOM)

NOTE - Changes from the top example of figures 4 and 5 are underlined.

Figure 7 - Effects of several changes in text aspects

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

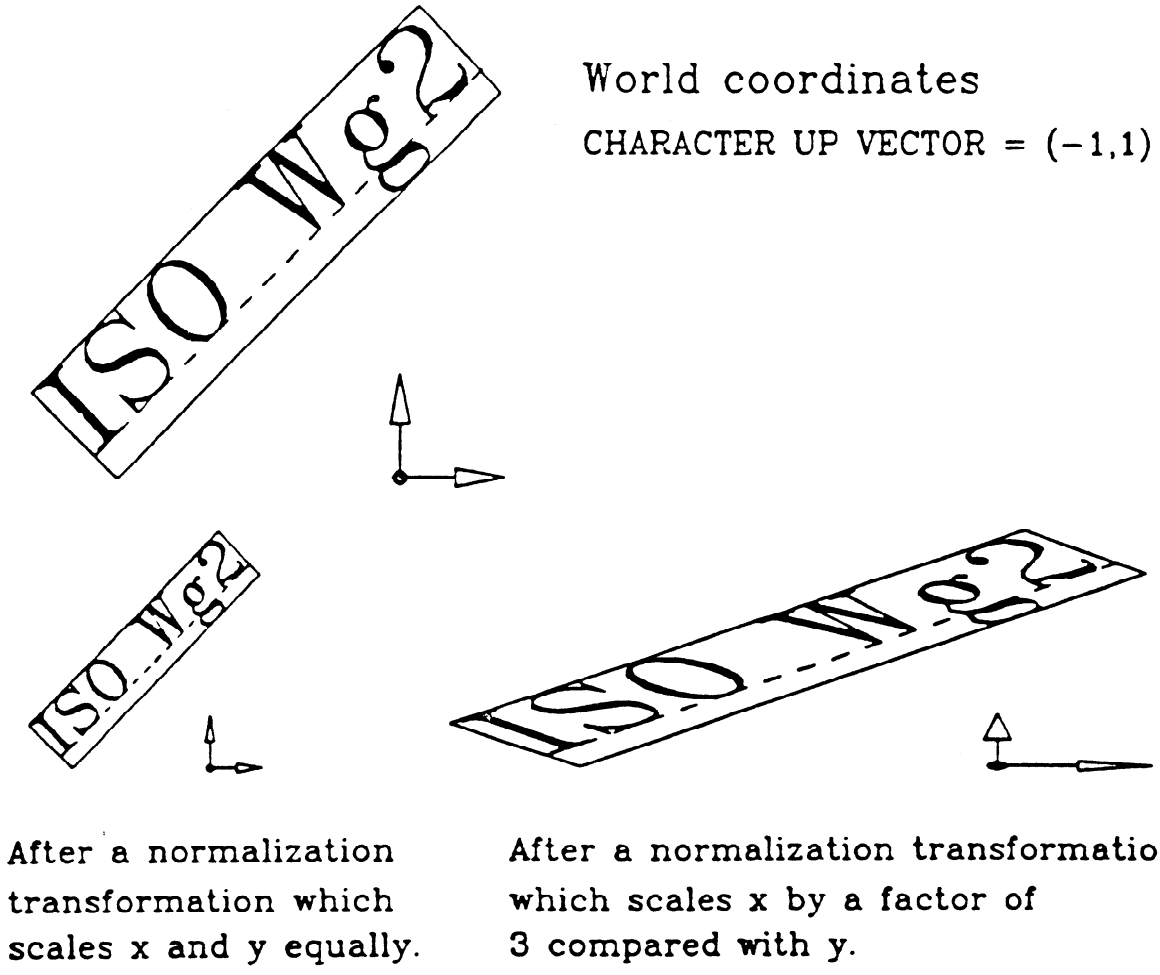
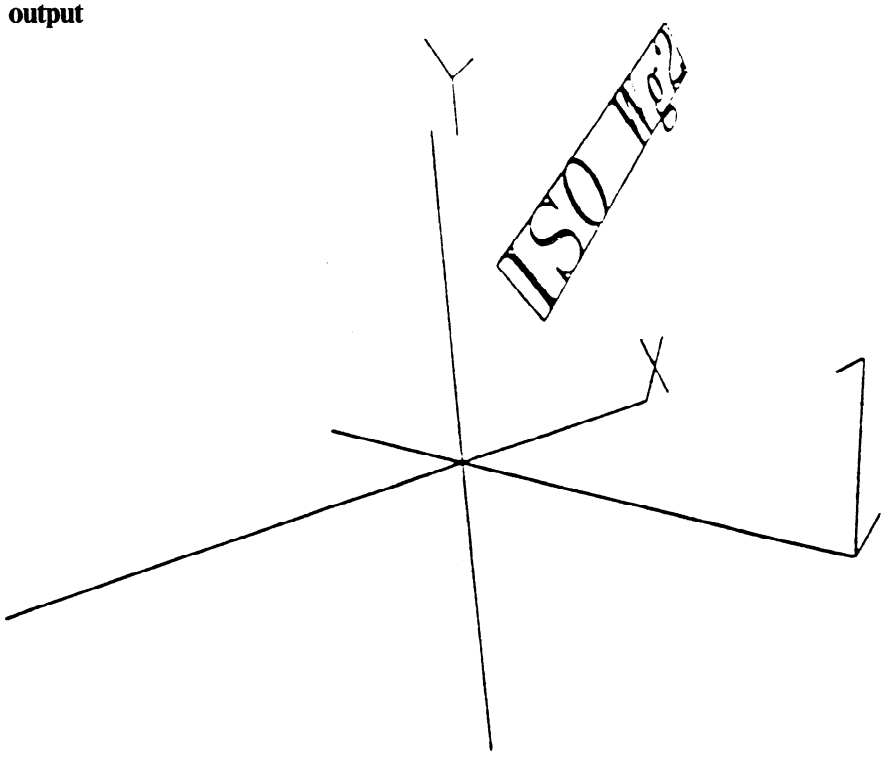


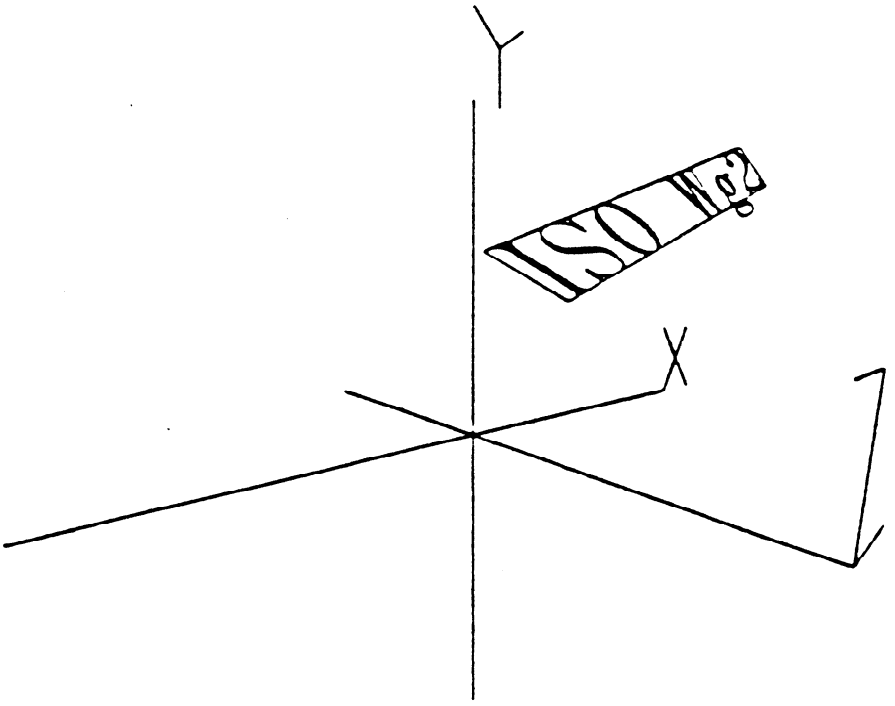
Figure 8 - Effects of different normalization transformations on text in STROKE precision

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988



world coordinates: character up vector = (-1,1)



After a normalization transformation which scales
x and z by a factor of 2 compared with y.

Figure 9 - Effects of different normalization transformations on text in STROKE precision

Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

The effect to be achieved is now described. A series of rectangles, aligned and positioned on the text plane by applying the current text attributes to the nominal character body extent rectangle, is passed through the viewing pipeline. The rectangle height vector has the length CHARACTER HEIGHT and is parallel to the actual character up vector (i.e. the 3D vector in WC that is equal to $chu_1 E_1 + chu_2 E_2$, where chu_1 and chu_2 are the components of the 2D CHARACTER UP VECTOR and E_1 and E_2 are unit vectors parallel to the positive X axis and Y axis in the text plane). The rectangle width vector has the length CHARACTER WIDTH and is parallel to the actual character base vector (i.e. the 3D vector in WC that is equal to $chb_1 E_1 + chb_2 E_2$, where chb_1 and chb_2 are the components of the 2D CHARACTER BASE VECTOR). The world coordinates of the four corners of the rectangle are transformed by the normalization transformation, by a segment transformation if within a segment, by a viewing transformation, and by the workstation transformation. The resulting quadrilateral can be used by the workstation character generator. Thus, the shape of individual characters can be transformed by a normalization transformation that scales x, y and z non-uniformly, by a segment transformation, by a viewing transformation and by a workstation transformation that scales z non-uniformly. Before the viewing transformation, the character body extent is a parallelogram and can be conveniently represented by the text position and two vectors to the two adjacent corners of the parallelogram. This parallelogram is stored in segments.

The resulting characters need to conform to the shape of the character extent quadrilateral exactly for STROKE precision. For STRING precision text, the character height and width vectors need to be evaluated only at the leading edge of the first character. For CHAR precision, the character height and width vectors need to be evaluated only at the leading edge of each character.

On the workstation, the height of a character is given by the length of the transformed height vector; the character up direction is given by the direction of the transformed height vector; the width of a character is given by the length of the transformed width vector multiplied by the font width to height ratio for the character and by the character expansion factor; the character base direction is given by the direction of the transformed width vector. The characters are arranged together in a text extent parallelogram, depending on the values of TEXT PATH and character spacing. The text extent parallelogram is then positioned according to the value of TEXT ALIGNMENT and the text position, contained in the definition of the TEXT primitive.

Figures 4 to 7 give examples of the effects of different values of text aspects. Figures 8 and 9 give examples of the effect of different normalization transformations on the displayed form of the text.

4.4.6 Text extent and concatenation

The text extent rectangle is the minimum rectangle which completely encloses the character bodies of the text string. This may be determined by INQUIRE TEXT EXTENT or INQUIRE TEXT EXTENT 3 which calculates the value of the text extent rectangle.

For font 1, the results are workstation independent. For other fonts, the results may be workstation dependent. STROKE precision text is assumed so that the information retrieved is exact.

4.4.7 Fill area attributes

Fill area has the geometric attributes PATTERN REFERENCE POINT and two PATTERN REFERENCE VECTORS. It also has two implicitly specified geometric attributes PATTERN WIDTH VECTOR and PATTERN HEIGHT VECTOR. These are implicitly specified by the function SET PATTERN SIZE. Like ordinary geometric attributes, their values are bound to the fill area primitives when the primitives are created and cannot be changed afterwards. These values are subject to the same transformations as the geometric data contained in the definition of the primitive. The usage of the fill area geometric attributes is described later in this sub-clause.

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988

The representation of fill area at the workstation is controlled by the FILL AREA INDEX, or the set of individually specified fill area attributes (FILL AREA INTERIOR STYLE, FILL AREA STYLE INDEX, and FILL AREA COLOUR INDEX) or some combination of the two, depending upon the values of the ASFs for fill area interior style, fill area style index, and fill area colour index. The FILL AREA INDEX is a pointer into the fill area bundle table, each entry of which contains values for the fill area interior style, fill area style index and fill area colour index.

The fill area interior style is used to determine in what style the area should be filled. It has the following values:

a) HOLLOW: No filling, but draw the bounding polyline, using the fill area colour index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF). The linetype and linewidth are implementation dependent.

b) SOLID: Fill the interior of the polygon using the fill area colour index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF).

c) PATTERN: Fill the interior of the polygon using the fill area style index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASF) as an index into the pattern table. In this context, the fill area style index is sometimes referred to as the pattern index.

d) HATCH: Fill the interior of the polygon using the fill area colour index and the fill area style index currently selected (either via the fill area bundle or individually, depending upon the corresponding ASFs). The fill area style index is used as a pointer into the list of hatch styles, in which case it is sometimes referred to as the hatch index.

For interior style PATTERN, the pattern is defined by the pattern representation, which specifies an array (DX×DY) of colour indices that are pointers into the colour table. The size and starting position of the pattern are determined by a pattern box in a local coordinate system on the fill area plane. The pattern box is defined as follows:

e) The PATTERN REFERENCE POINT and the PATTERN REFERENCE VECTORS are projected along a normal vector onto the fill area plane.

f) The origin of the box, which may be a parallelogram, is the projection of the PATTERN REFERENCE POINT.

g) The PATTERN WIDTH VECTOR is one side of the pattern box. Its direction is parallel to the projection of the first PATTERN REFERENCE VECTOR. Its length is equal to that of the SX component of the current PATTERN SIZE.

h) The PATTERN HEIGHT VECTOR is the adjacent side of the pattern box. Its direction is parallel to the projection of the second PATTERN REFERENCE VECTOR. Its length is equal to the SY component of the current PATTERN SIZE.

If the projected pattern box is degenerate, an implementation dependent technique determines the location and shape of the pattern box.

The pattern box is conceptually divided into a grid of DX×DY cells. The colour index array is associated with the cells as follows: the element (1,DY) is associated with the cell having the projected PATTERN REFERENCE POINT at one corner; elements with increasing first dimension are associated with successive cells in the direction of the PATTERN WIDTH VECTOR. Elements with decreasing second dimension are associated with successive cells in the direction of the PATTERN HEIGHT VECTOR. The pattern is mapped onto the fill area by replicating the transformed pattern box in directions parallel to its sides until the entire fill area is covered. The resulting

AVA Documents
Document Sharing Hub

Full document details: BS ISO 8805:1988