

INTERNATIONAL STANDARD

ISO/IEC
8806-4

First edition
1991-12-15

Information technology — Computer graphics — Graphical Kernel System for Three Dimensions (GKS-3D) language bindings —

Part 4: C

*Technologies de l'information — Infographie — Interfaces de langages
pour GKS (Graphical Kernel System) pour trois dimensions —*

Partie 4: C



Reference number
ISO/IEC 8806-4:1991(E)



Documents

Document Sharing Hub

Contents

1	Scope.....	1
2	Normative references.....	2
3	The C Language Binding of GKS-3D	3
3.1	Conformance	3
3.2	Functions versus Macros	3
3.3	Character Strings	3
3.4	Function Identifiers	3
3.5	Registration.....	3
3.6	Identifiers for Graphical Items.....	4
3.7	Return Values	4
3.8	Header Files.....	4
3.9	Memory Management.....	4
3.9.1	Functions which Return Simple Lists.....	5
3.9.2	Functions which Return Complex Data Structures	5
3.10	Error Handling.....	6
3.10.1	Application Supplied Error Handlers	6
3.10.2	Error Codes.....	7
3.10.3	C-specific GKS-3D errors	7
3.11	Colour Representations.....	7
3.12	Storage of Multi-dimensional Arrays	8
3.12.1	Storage of 2*3 Matrices.....	8
3.12.2	Storage of 3*4 Matrices.....	8
3.12.3	Storage of 4*4 Matrices.....	8
3.12.4	Storage of Colour Arrays.....	8
4	Tables.....	9
4.1	Abbreviation Policy in Construction of Identifiers.....	9
4.2	Table of Abbreviations Used.....	9
4.3	Function Names	13
4.3.1	List Ordered Alphabetically by Bound Name	13
4.3.2	List Ordered Alphabetically by GKS-3D Name	19
4.3.3	List Ordered Alphabetically by Bound Name within Level.....	25
5	Type Definitions	32
5.1	Mapping of GKS-3D data types	32
5.2	Environmental Type Definitions.....	32
5.3	Implementation Dependent Type Definitions	33
5.4	Implementation Independent Type Definitions	44
6	Macro Definitions	65
6.1	Function identifiers	65
6.2	Error Codes.....	68
6.3	Miscellaneous Macros	75
6.3.1	Linetypes.....	75
6.3.2	Marker Types.....	75
6.3.3	Colour Models	75
6.3.4	Prompt and Echo Types.....	75
6.3.5	Default Parameters of OPEN GKS.....	76
7	C GKS-3D Function Interface	77
7.1	Notational Conventions	77
7.2	Control Functions	77
7.3	Output Functions	79

© ISO/IEC 1991

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

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

Printed in Switzerland

7.4	Output Attribute Functions	81
7.4.1	Workstation Independent Primitive Attributes	81
7.4.2	Workstation Attributes	85
7.5	Transformation Functions	87
7.5.1	Normalization Transformation	87
7.5.2	View transformation functions	88
7.5.3	HLHSR functions	88
7.5.4	Workstation transformation	89
7.6	Segment Functions	89
7.6.1	Segment Manipulation Functions	89
7.6.2	Segment Attribute Functions	91
7.7	Input Functions	92
7.7.1	Initialization of Input Devices Functions	92
7.7.2	Setting the Mode of Input Devices Functions	95
7.7.3	Request Input Functions	96
7.7.4	Sample Input Functions	98
7.7.5	Event Input Functions	100
7.8	Metafile Functions	102
7.9	Inquiry Functions	102
7.9.1	Inquiry Functions for Operating State Value	102
7.9.2	Inquiry Functions for GKS-3D Description Table	103
7.9.3	Inquiry Functions for GKS-3D State List	103
7.9.4	Inquiry Functions for Workstation State List	112
7.9.5	Inquiry Functions for Workstation Description Table	124
7.9.6	Inquire functions for the Segment State List	135
7.9.7	Pixel Inquiries	136
7.9.8	Inquiry Functions for Error State List	137
7.10	Utility Functions	137
7.10.1	Utility Functions in GKS-3D	137
7.10.2	Binding Specific Utilities	138
7.11	Error Handling	139
A	Compiled GKS-3D/C Specification	141
B	Sample Programs	204
B.1	STAR Program	204
B.2	IRON Program	206
B.3	MAP Program	213
B.4	MANIPULATE Program	215
B.5	SHOW LINE Program	221
B.6	DODECA Program	226
C	Metafile Items	231
D	Short Function Identifiers	233
E	Memory Management	240
E.1	Introduction	240
E.2	Functions That Return Simple Lists	240
E.2.1	Operation of ginq_list_line_inds	240
E.3	Functions That Return Structured Data	243
E.3.1	Operation of gcreate_store	244
E.3.2	Operation of ginq_stroke_st and ginq_pat_rep	246
E.3.3	Operation of gdel_store	250
F	Function Lists	253
F.1	Alphabetic by GKS-3D Name	253
F.2	Alphabetic by Binding Name	259

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8806-4 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology*.

ISO/IEC 8806 consists of the following parts, under the general title

Information technology - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) language bindings -

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

Annexes A to F of this part of ISO/IEC 8806 are for information only.

Introduction

The Graphical Kernel System for Three Dimensions (GKS-3D) functional description is registered as ISO 8805 : 1988. As explained in the Scope and Field of Application of ISO 8805, that International Standard is specified in a language independent manner and needs to be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this part of ISO/IEC 8806 is to define a standard binding for the C computer programming language.

Some functions in ISO 8805 also appear in the Graphical Kernel System (GKS) functional description (ISO 7942). The binding of those functions in this part of ISO/IEC 8806 is the same as in ISO/IEC 8651-4, the C language binding of GKS.

This page intentionally left blank

Information technology – Computer graphics – Graphical Kernel System for Three Dimensions (GKS-3D) language bindings -

Part 4:

C

1 Scope

The Graphical Kernel System for Three Dimensions (GKS-3D), ISO 8805 : 1988 , specifies 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. This part of ISO/IEC 8806 specifies such a language dependent layer for the C language.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provision of this part of ISO/IEC 8806. At the time of publication, the editions indicated were valid. All standards are subject to revisions, and parties to agreements based on this part of ISO/IEC 8806 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7942:1985, *Information processing systems – Computer graphics – Graphical Kernel System (GKS) functional description*.

ISO/IEC 8651-4:1991, *Information technology – Computer graphics – Graphical Kernel System (GKS) language bindings - Part 4 : C*.

ISO 8805:1988, *Information processing systems – Computer graphics – Graphical Kernel System for Three Dimensions (GKS-3D) functional description*.

ISO/IEC 8806-1:1991, *Information processing systems – Computer graphics – Graphical Kernel System for three dimensions (GKS-3D) - language bindings - Part 1 : FORTRAN*.

ISO/IEC 9899:1990, *Programming languages - C*.

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

3 The C Language Binding of GKS-3D

The C language binding of GKS-3D shall be as described in clauses 3, 4, 5 and 6.

3.1 Conformance

This part of ISO/IEC 8806 incorporates the rules of conformance defined in the GKS-3D Standard (ISO 8805) for GKS-3D implementations, with those additional requirements specifically defined for C bindings in GKS-3D.

The following criteria shall determine conformance of an implementation to this part of ISO/IEC 8806:

In order to conform, an implementation of the C binding of GKS-3D shall implement a specific level of GKS-3D as specified in ISO 8805. It shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 8806 for that same level of GKS-3D and all lower levels and for a specific level of C.

Thus, for example, the syntax of the function names shall be precisely as specified in the binding and parameters shall be of the data types stated in the binding.

3.2 Functions versus Macros

An implementation may substitute macros for functions. However, the macros shall be designed so that side-effects work properly. In general, a macro cannot be used to replace the error handling function `gerr_hand`. See also 3.10.

3.3 Character Strings

The C language represents character strings as an array of characters terminated by the null character (i.e. `'\0'`). This means that the null character is not usable as a printable character.

3.4 Function Identifiers

The function names of GKS-3D are all mapped to C functions which begin with the letter `g`. Words and phrases used in the GKS-3D function names are often abbreviated in the representation and are always separated with the underscore character `"_"`. The set of such abbreviations is given in 4.2, and the resulting C function names are listed in 4.3. For example, the abbreviation for the GKS-3D function DELETE SEGMENT FROM WORKSTATION is `gdel_seg_ws`. `del`, `seg`, and `ws` are abbreviations for DELETE, SEGMENT, and WORKSTATION. The conjunctive FROM is mapped to the null string.

The C standard (ISO/IEC 9899) requires that compilers recognize internal identifiers which are distinct in at least 31 characters. That standard also requires that external identifiers (i.e. those seen by the linker) be recognized to a minimum of six characters, independent of case.

Implementations which run in environments where two distinct C internal identifiers would be equivalent, if they were both external identifiers, shall include a set of `#defines` in the header file which equate the long names to a set of short names. A possible set of short names for a compiler that accepts only eight characters for external definitions may be found in annex D.

3.5 Registration

ISO 8805 reserves certain value ranges for registration¹ as graphical items. The registered graphical items will be bound to the C programming language (and other programming languages). The registered item

¹) For the purpose of this part of ISO/IEC 8806 and according to the rules for the designation and operation of registration authorities in the ISO/IEC Directives, the ISO and IEC councils have designated the National Institute of Standards and Technology (Institute of Computer Sciences and Technology), A-266 Technology Building, Gaithersburg, MD 20899, USA to act as registration authority.

Registration**The C Language Binding of GKS-3D**

binding will be consistent with the binding presented in this part of ISO/IEC 8806.

3.6 Identifiers for Graphical Items

Generalized Drawing Primitives and Escape functions are referenced via identifiers. This part of ISO/IEC 8806 specifies the format of the identifiers but it does not specify the registration of the identifiers. The identifiers are used as arguments to the functions `ggdp`, `ggdp3`, and `gescape`.

An implementation may also represent GDPs and Escapes as separate functions, but this is not required.

There are two formats for these identifiers. One format is for registered GDPs and Escapes and the other format is for unregistered GDPs and Escapes.

The format for registered GDP identifiers is:

```
#define      GGDP_Rn      (n) /* 'n' is the registered GDP id. */
#define      GGDP3_Rn     (n) /* 'n' is the registered GDP3 id. */
```

The format for unregistered GDP identifiers is:

```
#define      GGDP_Un      (-n) /* 'n' is implementation dependent */
#define      GGDP3_Un     (-n) /* 'n' is the registered GDP3 id. */
```

The format for registered Escape function identifiers is:

```
#define      GESCAPE_Rn   (n) /* 'n' is the registered Escape id. */
```

The format for unregistered Escape function identifiers is:

```
#define      GESCAPE_Un   (-n) /* 'n' is implementation dependent */
```

3.7 Return Values

All GKS-3D/C functions return `void`.

3.8 Header Files

C provides a mechanism to allow external files to be included in a compilation. Clause 5 of this part of ISO/IEC 8806 describes the data types that shall be defined in the file `gks.h` which should be included in any application program that intends to use GKS-3D via the C binding.

This part of ISO/IEC 8806 uses the data type `size_t` (as a field in the data type `Gdata`). The type `size_t` is environment-dependent (e.g. `int`, `long int`, `unsigned int`) and is defined in the file `<stddef.h>`. Therefore the file `gks.h` shall also include the file `<stddef.h>`.

Additional implementation-dependent items may be placed in this file if needed. These items should start with the sentinel "G" or "g", as far as applicable.

The file `gks.h` shall also contain external declarations for all GKS-3D/C functions because they return `void`. For example, the declaration for the function `gopen_gks` would look like this:

```
extern void gopen_gks(char *err_file, size_t mem_units);
```

3.9 Memory Management

The application shall allocate the memory needed for the data returned by the implementation. In general, the application will allocate a C structure and pass a pointer to that structure to an inquiry routine, which will then place information into the structure. However, a number of inquiry functions return variable length data, the length of which is not known *a priori* by the application.

These functions fall into two classes. One class of functions returns a simple, homogeneous, list of items. For example, the function INQUIRE SET OF SEGMENT NAMES returns a list of the segment names in use. The other class returns complex, heterogeneous data structures. For example, the function INQUIRE LOCATOR DEVICE STATE returns the device state which includes a locator data record; the data record can contain arbitrarily complex implementation-defined data structures. The binding of these two classes of functions is described in detail below. Subclause 3.10 describes the errors that can be invoked during execution of functions which use the memory management policy.

The C Language Binding of GKS-3D

Memory Management

3.9.1 Functions which Return Simple Lists

Inquiry functions which return a list of items are bound such that the application can inquire about a portion of the list. This list is a subset of the implementation's internal list and is called the application's list. This allows the application to process the implementation's list in a piecewise manner rather than all at once.

The application allocates the memory for a list and passes that list to the implementation. The implementation places the results of the inquiry into the list. In order to support this policy of memory management, three additional parameters have been added to functions which return lists:

- a) `num_elems_appl_list`: An integer input parameter which is the length of the application's list. The value of `num_elems_appl_list` indicates the number of items (i.e. list elements) which will fit into the application list. A value of 0 is valid and allows the application to determine the size of the implementation's list (which is returned via `num_elems_impl_list`) without having the implementation return any of the elements of its list. If `num_elems_appl_list` is negative, `GE_APPL_LIST_LENGTH_LT_ZERO` is returned as the value of the error indicator parameter.
- b) `start_ind`: An integer input parameter which is an index into the implementation's list. (Index 0 is the first element of both the implementation's and application's list.) `start_ind` indicates the first item in the implementation's list that is copied into index 0 of the application's list. Items are copied sequentially from the implementation's list into the application's list until the application's list is full or there are no more items in the implementation's list. If `start_ind` is out of range, error `GE_START_IND_INVALID` is returned as the value of the error indicator parameter.
- c) `num_elems_impl_list`: An output parameter which is a pointer to an integer. The implementation stores into this parameter the number of items that are in the implementation's list.

In annex E, a possible underlying mechanism is described.

3.9.2 Functions which Return Complex Data Structures

The data returned by the `ESCAPE` function and the functions which return input device data records or pattern tables can be complex in structure. They cannot be represented by a simple list of items. It would be an onerous task for the application to have to allocate and prepare data structures for these routines. In order to facilitate this task of using these inquiry functions, the binding defines a new resource, called a *Store*, to manage the memory for these functions.

The *Store* resource is opaque to the application. The application does not know the structure of the *Store* or how it is implemented. The *Store* is defined as a `void *`. This part of ISO/IEC 8806 defines two new functions which create (in `CREATE STORE`, bound as `gcreate_store`) and delete (in `DELETE STORE`, bound as `del_store`) a *Store*.

A *Store* is used by the implementation to manage the memory needed by the functions which return complex data structures. Without specifying an implementation of a *Store*, it is safe to say that it will contain and control memory needed to hold the data returned by these functions and also contain some bookkeeping information about the contents and size of the memory.

The semantics of the *Store* resource provide two levels of memory management. The implementation is responsible for managing the memory at a low level because it uses, reuses, allocates and deallocates memory from the system in order to return information to the application. But the application is ultimately responsible for managing the memory at a high level because it creates and deletes *Stores*.

A *Store* is passed as a parameter to a function returning complex data structures. Another parameter to this function is a pointer to a pointer to a structure which defines the format of the returned data. The *Store* contains memory for the structure and any additional memory referenced by fields within the structure. The application accesses the returned data through its pointer to the structure. It does not use the *Store* to access the data.

A *Store* continues to hold the information from the function until the *Store* is deleted by the `DELETE STORE` function or until the *Store* is used as an argument to a subsequent function, which returns complex

Memory Management

The C Language Binding of GKS-3D

data structures. At that time, the old information is replaced with the new. Thus multiple calls to functions overwrite the contents of a *Store*. A *Store* only contains the results of the last function.

This part of ISO/IEC 8806 defines two new errors that can occur when using or creating a *Store*; these errors are described in 3.10.3. For most functions using a *Store*, these and other errors are returned via the "error indicator" parameter. However, the function ESCAPE does not have an error indicator parameter. For this function, the error reporting mechanism is used when an error is encountered. For this function, the implementation shall, in addition to reporting the error, set the pointer to the returned data to NULL when an error occurs. See the binding of these functions for more information.

The definitions for the functions CREATE STORE and DELETE STORE follow:

CREATE STORE	GKOP, WSOP, WSAC, SGOP	L0a
---------------------	-------------------------------	------------

Parameters:

Out	error indicator	I
Out	store	STORE

Effect: Creates a *Store* and returns a handle to it in the output parameter *store*. If the *Store* cannot be created, the *store* parameter is set to NULL and the error indicator is set to one of the following error values:

8	GKS-3D not in proper state; GKS-3D shall be in one of the states GKOP, WSOP, WSAC or SGOP
2203	Error while allocating Store

Errors: None.

DELETE STORE	GKOP, WSOP, WSAC, SGOP	L0a
---------------------	-------------------------------	------------

Parameters:

Out	error indicator	I
In/Out	store	STORE

Effect: Deletes the *Store* and all internal resources associated with it. If there is not an error, the parameter *store* will be set to NULL to signify that it is no longer valid. If an error is detected, the error indicator is set to one of the following values:

8	GKS-3D not in proper state; GKS-3D shall be in one of the states GKOP, WSOP, WSAC or SGOP
---	---

Errors: None.

In 7.10.2, the C specification of these functions is given. In annex E, a possible underlying mechanism is illustrated.

3.10 Error Handling

3.10.1 Application Supplied Error Handlers

User-defined error handlers shall accept the same arguments as the standard error handler. The user-defined error handler is specified by the utility function (see also 7.10.2)

SET ERROR HANDLER**GKCL, GKOP, WSOP, WSAC, SGOP****L0a***Parameters:*

In	New error handling function	Function
Out	Old error handling function	Function

Effect: Sets the GKS-3D error handling function to *New error handling function* and returns the current error handling function to *Old error handling function*.

Errors: None.

Application defined error handling functions accept the same arguments as the standard error handler. They may invoke the standard error logging function ERROR LOGGING.

ISO 8805 defines the initial error handling function to be ERROR HANDLING, that is, the value of the parameter *Old error handling function* points to ERROR HANDLING, when SET ERROR HANDLER is called for the first time.

When the application changes the error handling function, the implementation will invoke the new function when an error is detected. If the application calls the default error handling function ERROR HANDLING, ERROR HANDLING will always call the function ERROR LOGGING.

If *New error handler* is not a valid pointer, the error handling will automatically be done by the standard error handler ERROR HANDLING.

User-defined error handlers may invoke the standard error logging function ERROR LOGGING.

3.10.2 Error Codes

Hard coding numbers into a program decreases its maintainability. Therefore, this part of ISO/IEC 8806 defines a set of constants for the GKS-3D error numbers. Each error constant begins with the characters GE_. See also 6.2 for the error macros.

3.10.3 C-specific GKS-3D errors

This part of ISO/IEC 8806 knows some specific error messages. This section gives the messages and the error macros.

Value	Message
2200	Start index out of range Is issued when the start index is less than zero or larger than the last element in the implementation's list
2201	Length of application list is negative Is issued when the length of the application's list is less than zero
2202	Enumeration type out of range Is issued when a parameter value whose type is an enumeration is out range of the enumeration
2203	Error while allocating Store Is issued when an error is detected during CREATE STORE
2204	Error while allocating memory for Store Is issued when a function using a <i>Store</i> is unable to allocate memory for the <i>Store</i>

3.11 Colour Representations

GKS-3D defines 4 colour models (RGB, CIE L*u*v* 1976, HLS and HSV) of which RGB and CIE L*u*v* are mandatory. For each of these models, a colour specification is defined (*Grgb*, *Gcieluv*, *Ghsv*, *Ghls*). The colour representation is defined in GKS-3D/C as

Colour Representations

The C Language Binding of GKS-3D

```

typedef union {
    Grgb      rgb;          /* Red Green Blue colour specification */
    Gcieluv   cielu;        /* CIE L*u*v* 1976 colr. spec. */
    Ghls      hls;          /* Hue Luminance Saturation colr. spec. */
    Ghsv      hsv;          /* Hue Saturation Value colr. spec. */
    Gdata      impl_dep;    /* impl. dep. colr. spec. */
} Gcolr_rep;

```

which guarantees extensibility of GKS-3D/C for other colour models.

3.12 Storage of Multi-dimensional Arrays

3.12.1 Storage of 2*3 Matrices

The entries of `Gtran_matrix` data types shall be stored such that the segment transformation is defined by

$$\begin{aligned} \text{Tp.x} &= \text{mat}[0,0] * \text{p.x} + \text{mat}[0,1] * \text{p.y} + \text{mat}[0,2]; \\ \text{Tp.y} &= \text{mat}[1,0] * \text{p.x} + \text{mat}[1,1] * \text{p.y} + \text{mat}[1,2]; \end{aligned}$$

where `p` is a 2D point, `Tp` its transformation and `mat` is of type `Gtran_matrix`.

3.12.2 Storage of 3*4 Matrices

The entries of `Gtran_matrix3` data types shall be stored such that the segment transformation is defined by

$$\begin{aligned} \text{Tp.x} &= \text{mat}[0,0] * \text{p.x} + \text{mat}[0,1] * \text{p.y} + \text{mat}[0,2] * \text{p.z} + \text{mat}[0,3]; \\ \text{Tp.y} &= \text{mat}[1,0] * \text{p.x} + \text{mat}[1,1] * \text{p.y} + \text{mat}[1,2] * \text{p.z} + \text{mat}[1,3]; \\ \text{Tp.z} &= \text{mat}[2,0] * \text{p.x} + \text{mat}[2,1] * \text{p.y} + \text{mat}[2,2] * \text{p.z} + \text{mat}[2,3]; \end{aligned}$$

where `p` is a 3D point, `Tp` its transformation and `mat` is of type `Gtran_matrix3`.

3.12.3 Storage of 4*4 Matrices

The entries of `Gmatrix3` data types shall be stored such that the projection transformation is defined by

$$\begin{aligned} \text{Tp.x} &= x' / w'; \\ \text{Tp.y} &= y' / w'; \\ \text{Tp.z} &= z' / w'; \\ x' &= \text{mat}[0,0] * \text{p.x} + \text{mat}[0,1] * \text{p.y} + \text{mat}[0,2] * \text{p.z} + \text{mat}[0,3]; \\ y' &= \text{mat}[1,0] * \text{p.x} + \text{mat}[1,1] * \text{p.y} + \text{mat}[1,2] * \text{p.z} + \text{mat}[1,3]; \\ z' &= \text{mat}[2,0] * \text{p.x} + \text{mat}[2,1] * \text{p.y} + \text{mat}[2,2] * \text{p.z} + \text{mat}[2,3]; \\ w' &= \text{mat}[3,0] * \text{p.x} + \text{mat}[3,1] * \text{p.y} + \text{mat}[3,2] * \text{p.z} + \text{mat}[3,3]; \end{aligned}$$

where `p` is a 3D point, `Tp` its projection transformation and `mat` is of type `Gmatrix3`.

3.12.4 Storage of Colour Arrays

The entries of `Gpat_rep` data types shall be stored such that the colour index at the (i,j)-th entry is given by

$$\begin{aligned} \text{Colr_ind}_{i,j} &= \text{colr_rect.colr_array}[i + \text{DX} * j]; \\ i &= 0, \dots, \text{DX} - 1; & j &= 0, \dots, \text{DY} - 1; \\ \text{DX} &= \text{colr_rect.dims.size_x}; & \text{DY} &= \text{colr_rect.dims.size_y}; \end{aligned}$$

where `colr_rect` is of type `Gpat_rep`.

4 Tables

4.1 Abbreviation Policy in Construction of Identifiers

In the construction of the several data types, function names, etc., the following policy is applied:

- 1) All identifiers in the C binding are abbreviated using the same abbreviations for every component and using underscores to denote blanks;
- 2) The plural of an expression is constructed by adding an "s" after its abbreviation; so, for example, "vector" is abbreviated to "vec" and "vectors" is abbreviated to "vecs"; if an expression is mapped to NULL, so will be its plural;
- 3) Digits are also preceded by underscores, except for the 3, if the 3D extension is denoted;
- 4) The words POLYLINE, POLYMARKER and FILL AREA (SET) are not abbreviated in the output primitive function names; in all other cases they are abbreviated using the list in 4.2;
- 5) Construction of GKS-3D/C identifiers:
 - a) Function names:
"g" (lower case) followed by abbreviated function name in lower case;
 - b) Data types:
"G" (upper case) followed by abbreviated data type in lower case;
 - c) Fields of data types; the following refinements are used: "redundant" (words in the field name that are identical to those in the structure name) parts are omitted, if the context allows this; thus the colour index in the field of `Gxxx_bundle` is abbreviated to `colr_ind`, because the context makes clear which colour index is used;
 - d) Function macros:
"Gfn_" followed by abbreviation of function name;
 - e) GKSM item types:
"Gksm_" followed by abbreviation of item name;
 - f) Error macros:
"GE_" followed by some abbreviated expression;
 - g) Fields of enumeration types:
"G" (upper case) followed by a prefix followed by an abbreviation of the field name; this prefix is constant for each enumeration field; all the fields are in upper case;

4.2 Table of Abbreviations Used

In this table, only words which are abbreviated are listed. They are used for

function names;
data types;
fields of data types;
error macros;
GKSM item types.

The word "NULL" denotes those words which are deleted completely when forming function names or data types.

Word or Phrase	Abbreviation
accepted	NULL

Table of Abbreviations Used

Tables

accumulate	accum
action	NULL
actual	act
addition	add
alignment	align
allocate	alloc
and	NULL
application	appl
arithmetic	arith
as soon as possible	asap
aspect source flag	asf
associate(d)	assoc
at some time	asti
at	NULL
attribute	attr
availability	avail
available	avail
before the next interaction globally	bnig
before the next interaction locally	bnil
between	NULL
buffer	buf
cannot	cant
category	cat
centre	ctr
character	char
CIE $L^*u^*v^*$ (colour model)	cieluv
classification	class
clipping	clip
collinear	col
colour	colr
concatenation	concat
conditional(ly)	cond
connection	conn
control	ctrl
coordinate	coord
corner	cnr
current	cur
dashed	dash
default	def
deferral	defer
delete	del
deletion	del
detectability	det
detectable	det
device coordinate(s)	dc
device(s)	NULL
digital	digit
dimension	dim
direction	dir
display	disp
distance	NULL
dotted	dot

Tables

Table of Abbreviations Used

duplicate	dup
dynamic	dyn
equal	eq
error	err
evaluate	eval
expansion	expan
facility	fac
factor	NULL
fill area	fill
from	NULL
function	func
generalized drawing primitive	gdp
gks closed	gkcl
gks open	gkop
gksm	NULL
graphical	graph
handling	hand
height	ht
hidden line/hidden surface removal	hlhsr
highlighted	highl
highlighting	highl
horizontal	hor
hue lightness saturation (colour model)	hls
hue saturation value (colour model)	hsv
identifier	id
immediate(ly)	imm
implicit regeneration	irg
implicit	impl
in use	NULL
in	NULL
index	ind
indicator	ind
individual	indiv
initial	init
initialize	init
input/output	io
input	in
inquire	inq
integer	int
interior	int
invalid	inval
invisible	invis
length	NULL
less than or equal	le
less than	lt
level x	lx (with x = 0a, 0b, ...)
library	lib
locator	loc
logical	NULL
luminance	lum
mapping	map
maximum	max

Table of Abbreviations Used

Tables

memory	mem
metafile input	mi
metafile output	mo
minimum	min
modification	mod
monochrome	monochr
necessary	nec
nominal	nom
normal	norm
normalization	norm
normalized device coordinate(s)	ndc
number	num
of	NULL
on	NULL
operating	op
orientation	ori
output and input	outin
output	out
overflow	overf
parallel(ogram)	paral
pattern	pat
pending	pend
perspective	perspect
pointer	ptr
polyline	line
polymarker	marker
position	pos
precision	prec
predefined	pred
presence	pres
primitive	prim
priority	pri
projection	proj
prompt and echo type	pet
prompt	pr
queue	NULL
record	NULL
rectangle	rect
red green blue (colour model)	rgb
reference	ref
regeneration	regen
registered	r
relative	rel
representation	rep
request(ed)	req
rubber	rub
saturation	satur
scale factor	NULL
segment open	sgop
segment	seg
select	sel
simultaneous(ly)	simult

Tables**Table of Abbreviations Used**

spacing	space
specification	specif
specified	specif
state	st
storage	store
supported	NULL
suppressed	suppr
to	NULL
tracking	track
transform(ation)	tran
unavailable	unavail
undefined	undef
undetectable	undet
unregistered	u
update	upd
used	NULL
valuator	val
value	NULL
vector	vec
vertical	vert
viewport	vp
visibility	vis
visible	vis
volume	vol
which	NULL
window	win
with	NULL
workstation active	wsac
workstation independent segment store	wiss
workstation open	wsop
workstation	ws
world coordinate	wc

4.3 Function Names**4.3.1 List Ordered Alphabetically by Bound Name**

gaccum_tran_matrix	ACCUMULATE TRANSFORMATION MATRIX
gaccum_tran_matrix3	ACCUMULATE TRANSFORMATION MATRIX 3
gactivate_ws	ACTIVATE WORKSTATION
gassoc_seg_ws	ASSOCIATE SEGMENT WITH WORKSTATION
gawait_event	AWAIT EVENT
gcell_array	CELL ARRAY
gcell_array3	CELL ARRAY 3
gclear_ws	CLEAR WORKSTATION
gclose_gks	CLOSE GKS
gclose_seg	CLOSE SEGMENT
gclose_ws	CLOSE WORKSTATION
gcopy_seg_ws	COPY SEGMENT FROM WORKSTATION
gcreate_seg	CREATE SEGMENT
gcreate_store	CREATE STORE (GKS-3D/C only)
gdeactivate_ws	DEACTIVATE WORKSTATION

Function Names

Tables

gdel_seg	DELETE SEGMENT
gdel_seg_ws	DELETE SEGMENT FROM WORKSTATION
gdel_store	DELETE STORE (GKS-3D/C only)
gemergency_close_gks	EMERGENCY CLOSE GKS
gerr_hand	ERROR HANDLING
gerr_log	ERROR LOGGING
gescape	ESCAPE
geval_tran_matrix	EVALUATE TRANSFORMATION MATRIX
geval_tran_matrix3	EVALUATE TRANSFORMATION MATRIX 3
geval_view_map_matrix3	EVALUATE VIEW MAPPING MATRIX 3
geval_view_ori_matrix3	EVALUATE VIEW ORIENTATION MATRIX 3
gfill_area	FILL AREA
gfill_area3	FILL AREA 3
gfill_area_set	FILL AREA SET
gfill_area_set3	FILL AREA SET 3
gflush_events	FLUSH DEVICE EVENTS
ggdp	GENERALIZED DRAWING PRIMITIVE
ggdp3	GENERALIZED DRAWING PRIMITIVE 3
gget_choice	GET CHOICE
gget_item_type	GET ITEM TYPE FROM GKSM
gget_loc	GET LOCATOR
gget_loc3	GET LOCATOR 3
gget_pick	GET PICK
gget_string	GET STRING
gget_stroke	GET STROKE
gget_stroke3	GET STROKE 3
gget_val	GET VALUATOR
ginit_choice	INITIALIZE CHOICE
ginit_choice3	INITIALIZE CHOICE 3
ginit_loc	INITIALIZE LOCATOR
ginit_loc3	INITIALIZE LOCATOR 3
ginit_pick	INITIALIZE PICK
ginit_pick3	INITIALIZE PICK 3
ginit_string	INITIALIZE STRING
ginit_string3	INITIALIZE STRING 3
ginit_stroke	INITIALIZE STROKE
ginit_stroke3	INITIALIZE STROKE 3
ginit_val	INITIALIZE VALUATOR
ginit_val3	INITIALIZE VALUATOR 3
ginq_asfs	INQUIRE ASPECT SOURCE FLAGS
ginq_asfs3	INQUIRE ASPECT SOURCE FLAGS 3
ginq_char_base_vec	INQUIRE CHARACTER BASE VECTOR
ginq_char_expan	INQUIRE CHARACTER EXPANSION FACTOR
ginq_char_ht	INQUIRE CHARACTER HEIGHT
ginq_char_space	INQUIRE CHARACTER SPACING
ginq_char_up_vec	INQUIRE CHARACTER UP VECTOR
ginq_char_width	INQUIRE CHARACTER WIDTH
ginq_choice_st	INQUIRE CHOICE DEVICE STATE
ginq_choice_st3	INQUIRE CHOICE DEVICE STATE 3
ginq_clip	INQUIRE CLIPPING
ginq_clip3	INQUIRE CLIPPING 3
ginq_colr_fac	INQUIRE COLOUR FACILITIES

Tables

Function Names

ging_colr_model	INQUIRE COLOUR MODEL
ging_colr_model_fac	INQUIRE COLOUR MODEL FACILITIES
ging_colr_rep	INQUIRE COLOUR REPRESENTATION
ging_cur_hlshr_id	INQUIRE CURRENT HLHSR IDENTIFIER VALUE
ging_cur_indiv_attr	INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES
ging_cur_indiv_attr3	INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES 3
ging_cur_norm_tran_num	INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER
ging_cur_pick_id	INQUIRE CURRENT PICK IDENTIFIER VALUE
ging_cur_prim_attr	INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES
ging_cur_prim_attr3	INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES 3
ging_def_choice_data	INQUIRE DEFAULT CHOICE DEVICE DATA
ging_def_choice_data3	INQUIRE DEFAULT CHOICE DEVICE DATA 3
ging_def_defer_sts	INQUIRE DEFAULT DEFERRAL STATE VALUES
ging_def_loc_data	INQUIRE DEFAULT LOCATOR DEVICE DATA
ging_def_loc_data3	INQUIRE DEFAULT LOCATOR DEVICE DATA 3
ging_def_pick_data	INQUIRE DEFAULT PICK DEVICE DATA
ging_def_pick_data3	INQUIRE DEFAULT PICK DEVICE DATA 3
ging_def_string_data	INQUIRE DEFAULT STRING DEVICE DATA
ging_def_string_data3	INQUIRE DEFAULT STRING DEVICE DATA 3
ging_def_stroke_data	INQUIRE DEFAULT STROKE DEVICE DATA
ging_def_stroke_data3	INQUIRE DEFAULT STROKE DEVICE DATA 3
ging_def_val_data	INQUIRE DEFAULT VALUATOR DEVICE DATA
ging_def_val_data3	INQUIRE DEFAULT VALUATOR DEVICE DATA 3
ging_disp_space_size	INQUIRE DISPLAY SPACE SIZE
ging_disp_space_size3	INQUIRE DISPLAY SPACE SIZE 3
ging_dyn_mod_seg_attr	INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES
ging_dyn_mod_ws_attr	INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES
ging_dyn_mod_ws_attr3	INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3
ging_edge_colr_ind	INQUIRE EDGE COLOUR INDEX
ging_edge_fac	INQUIRE EDGE FACILITIES
ging_edge_flag	INQUIRE EDGE FLAG
ging_edge_ind	INQUIRE EDGE INDEX
ging_edge_rep	INQUIRE EDGE REPRESENTATION
ging_edgetype	INQUIRE EDGETYPE
ging_edgewidth	INQUIRE EDGEWIDTH SCALE FACTOR
ging_fill_colr_ind	INQUIRE FILL AREA COLOUR INDEX
ging_fill_fac	INQUIRE FILL AREA FACILITIES
ging_fill_ind	INQUIRE FILL AREA INDEX
ging_fill_int_style	INQUIRE FILL AREA INTERIOR STYLE
ging_fill_rep	INQUIRE FILL AREA REPRESENTATION
ging_fill_style_ind	INQUIRE FILL AREA STYLE INDEX
ging_gdp	INQUIRE GENERALIZED DRAWING PRIMITIVE
ging_gdp3	INQUIRE GENERALIZED DRAWING PRIMITIVE 3
ging_hlshr_id_fac	INQUIRE HLHSR IDENTIFIER FACILITIES
ging_hlshr_mode	INQUIRE HLHSR MODE
ging_hlshr_mode_fac	INQUIRE HLHSR MODE FACILITIES
ging_in_overf	INQUIRE INPUT QUEUE OVERFLOW
ging_level_gks	INQUIRE LEVEL OF GKS
ging_line_colr_ind	INQUIRE POLYLINE COLOUR INDEX
ging_line_fac	INQUIRE POLYLINE FACILITIES
ging_line_ind	INQUIRE POLYLINE INDEX
ging_line_rep	INQUIRE POLYLINE REPRESENTATION

Function Names

Tables

<code>ginq_linetype</code>	INQUIRE LINETYPE
<code>ginq_linewidth</code>	INQUIRE LINEWIDTH SCALE FACTOR
<code>ginq_list_avail_gdps</code>	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES
<code>ginq_list_avail_gdps3</code>	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3
<code>ginq_list_avail_ws_types</code>	INQUIRE LIST OF AVAILABLE WORKSTATION TYPES
<code>ginq_list_colr_inds</code>	INQUIRE LIST OF COLOUR INDICES
<code>ginq_list_edge_inds</code>	INQUIRE LIST OF EDGE INDICES
<code>ginq_list_fill_inds</code>	INQUIRE LIST OF FILL AREA INDICES
<code>ginq_list_line_inds</code>	INQUIRE LIST OF POLYLINE INDICES
<code>ginq_list_marker_inds</code>	INQUIRE LIST OF POLYMARKER INDICES
<code>ginq_list_norm_tran_nums</code>	INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS
<code>ginq_list_pat_inds</code>	INQUIRE LIST OF PATTERN INDICES
<code>ginq_list_text_inds</code>	INQUIRE LIST OF TEXT INDICES
<code>ginq_list_view_inds</code>	INQUIRE LIST OF VIEW INDICES
<code>ginq_loc_st</code>	INQUIRE LOCATOR DEVICE STATE
<code>ginq_loc_st3</code>	INQUIRE LOCATOR DEVICE STATE 3
<code>ginq_marker_colr_ind</code>	INQUIRE POLYMARKER COLOUR INDEX
<code>ginq_marker_fac</code>	INQUIRE POLYMARKER FACILITIES
<code>ginq_marker_ind</code>	INQUIRE POLYMARKER INDEX
<code>ginq_marker_rep</code>	INQUIRE POLYMARKER REPRESENTATION
<code>ginq_marker_size</code>	INQUIRE MARKER SIZE SCALE FACTOR
<code>ginq_marker_type</code>	INQUIRE MARKER TYPE
<code>ginq_max_norm_tran_num</code>	INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER
<code>ginq_max_ws_st_tables</code>	INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES
<code>ginq_max_ws_st_tables3</code>	INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3
<code>ginq_more_simult_events</code>	INQUIRE MORE SIMULTANEOUS EVENTS
<code>ginq_name_open_seg</code>	INQUIRE NAME OF OPEN SEGMENT
<code>ginq_norm_tran</code>	INQUIRE NORMALIZATION TRANSFORMATION
<code>ginq_norm_tran3</code>	INQUIRE NORMALIZATION TRANSFORMATION 3
<code>ginq_num_avail_in</code>	INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES
<code>ginq_num_seg_pris</code>	INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED
<code>ginq_op_st</code>	INQUIRE OPERATING STATE VALUE
<code>ginq_pat_fac</code>	INQUIRE PATTERN FACILITIES
<code>ginq_pat_ht_vec</code>	INQUIRE PATTERN HEIGHT VECTOR
<code>ginq_pat_ref_point</code>	INQUIRE PATTERN REFERENCE POINT
<code>ginq_pat_ref_point_vecs</code>	INQUIRE PATTERN REFERENCE POINT AND VECTORS
<code>ginq_pat_rep</code>	INQUIRE PATTERN REPRESENTATION
<code>ginq_pat_width_vec</code>	INQUIRE PATTERN WIDTH VECTOR
<code>ginq_pick_st</code>	INQUIRE PICK DEVICE STATE
<code>ginq_pick_st3</code>	INQUIRE PICK DEVICE STATE 3
<code>ginq_pixel</code>	INQUIRE PIXEL
<code>ginq_pixel_array</code>	INQUIRE PIXEL ARRAY
<code>ginq_pixel_array_dims</code>	INQUIRE PIXEL ARRAY DIMENSIONS
<code>ginq_pred_colr_rep</code>	INQUIRE PREDEFINED COLOUR REPRESENTATION
<code>ginq_pred_edge_rep</code>	INQUIRE PREDEFINED EDGE REPRESENTATION
<code>ginq_pred_fill_rep</code>	INQUIRE PREDEFINED FILL AREA REPRESENTATION
<code>ginq_pred_line_rep</code>	INQUIRE PREDEFINED POLYLINE REPRESENTATION
<code>ginq_pred_marker_rep</code>	INQUIRE PREDEFINED POLYMARKER REPRESENTATION
<code>ginq_pred_pat_rep</code>	INQUIRE PREDEFINED PATTERN REPRESENTATION
<code>ginq_pred_text_rep</code>	INQUIRE PREDEFINED TEXT REPRESENTATION
<code>ginq_pred_view_rep</code>	INQUIRE PREDEFINED VIEW REPRESENTATION
<code>ginq_seg_attrs</code>	INQUIRE SEGMENT ATTRIBUTES

Tables

Function Names

ginq_seg_attrs3	INQUIRE SEGMENT ATTRIBUTES 3
ginq_set_active_wss	INQUIRE SET OF ACTIVE WORKSTATIONS
ginq_set_assoc_wss	INQUIRE SET OF ASSOCIATED WORKSTATIONS
ginq_set_open_wss	INQUIRE SET OF OPEN WORKSTATIONS
ginq_set_seg_names	INQUIRE SET OF SEGMENT NAMES IN USE
ginq_set_seg_names_ws	INQUIRE SET OF SEGMENT NAMES ON WORKSTATION
ginq_string_st	INQUIRE STRING DEVICE STATE
ginq_string_st3	INQUIRE STRING DEVICE STATE 3
ginq_stroke_st	INQUIRE STROKE DEVICE STATE
ginq_stroke_st3	INQUIRE STROKE DEVICE STATE 3
ginq_text_align	INQUIRE TEXT ALIGNMENT
ginq_text_colr_ind	INQUIRE TEXT COLOUR INDEX
ginq_text_extent	INQUIRE TEXT EXTENT
ginq_text_extent3	INQUIRE TEXT EXTENT 3
ginq_text_facfs	INQUIRE TEXT FACILITIES
ginq_text_font_prec	INQUIRE TEXT FONT AND PRECISION
ginq_text_ind	INQUIRE TEXT INDEX
ginq_text_path	INQUIRE TEXT PATH
ginq_text_rep	INQUIRE TEXT REPRESENTATION
ginq_val_st	INQUIRE VALUATOR DEVICE STATE
ginq_val_st3	INQUIRE VALUATOR DEVICE STATE 3
ginq_view_facfs	INQUIRE VIEW FACILITIES
ginq_view_ind	INQUIRE VIEW INDEX
ginq_view_rep3	INQUIRE VIEW REPRESENTATION 3
ginq_ws_cat	INQUIRE WORKSTATION CATEGORY
ginq_ws_class	INQUIRE WORKSTATION CLASSIFICATION
ginq_ws_conn_type	INQUIRE WORKSTATION CONNECTION AND TYPE
ginq_ws_defer_upd_sts	INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES
ginq_ws_max_nums	INQUIRE WORKSTATION MAXIMUM NUMBERS
ginq_ws_st	INQUIRE WORKSTATION STATE
ginq_ws_tran	INQUIRE WORKSTATION TRANSFORMATION
ginq_ws_tran3	INQUIRE WORKSTATION TRANSFORMATION 3
ginsert_seg	INSERT SEGMENT
ginsert_seg3	INSERT SEGMENT 3
ginterpret_item	INTERPRET ITEM
gmessage	MESSAGE
gopen_gks	OPEN GKS
gopen_ws	OPEN WORKSTATION
gpolyline	POLYLINE
gpolyline3	POLYLINE 3
gpolymarker	POLYMARKER
gpolymarker3	POLYMARKER 3
gread_item	READ ITEM FROM GKSM
gredraw_all_segs_ws	REDRAW ALL SEGMENTS ON WORKSTATION
grename_seg	RENAME SEGMENT
greq_choice	REQUEST CHOICE
greq_loc	REQUEST LOCATOR
greq_loc3	REQUEST LOCATOR 3
greq_pick	REQUEST PICK
greq_string	REQUEST STRING
greq_stroke	REQUEST STROKE
greq_stroke3	REQUEST STROKE 3

Function Names

Tables

greq_val	REQUEST VALUATOR
gsample_choice	SAMPLE CHOICE
gsample_loc	SAMPLE LOCATOR
gsample_loc3	SAMPLE LOCATOR 3
gsample_pick	SAMPLE PICK
gsample_string	SAMPLE STRING
gsample_stroke	SAMPLE STROKE
gsample_stroke3	SAMPLE STROKE 3
gsample_val	SAMPLE VALUATOR
gsel_norm_tran	SELECT NORMALIZATION TRANSFORMATION
gset_asfs	SET ASPECT SOURCE FLAGS
gset_asfs3	SET ASPECT SOURCE FLAGS
gset_char_expan	SET CHARACTER EXPANSION FACTOR
gset_char_ht	SET CHARACTER HEIGHT
gset_char_space	SET CHARACTER SPACING
gset_char_up_vec	SET CHARACTER UP VECTOR
gset_choice_mode	SET CHOICE MODE
gset_clip_ind	SET CLIPPING INDICATOR
gset_colr_model	SET COLOUR MODEL
gset_colr_rep	SET COLOUR REPRESENTATION
gset_defer_st	SET DEFERRAL STATE
gset_det	SET DETECTABILITY
gset_edge_colr_ind	SET EDGE COLOUR INDEX
gset_edge_flag	SET EDGE FLAG
gset_edge_ind	SET EDGE INDEX
gset_edge_rep	SET EDGE REPRESENTATION
gset_edgetype	SET EDGETYPE
gset_edgewidth	SET EDGEWIDTH SCALE FACTOR
gset_err_hand	SET ERROR HANDLER (GKS-3D/C only)
gset_fill_colr_ind	SET FILL AREA COLOUR INDEX
gset_fill_ind	SET FILL AREA INDEX
gset_fill_int_style	SET FILL AREA INTERIOR STYLE
gset_fill_rep	SET FILL AREA REPRESENTATION
gset_fill_style_ind	SET FILL AREA STYLE INDEX
gset_highl	SET HIGHLIGHTING
gset_hlhr_id	SET HLHR IDENTIFIER
gset_hlhr_mode	SET HLHR MODE
gset_line_colr_ind	SET POLYLINE COLOUR INDEX
gset_line_ind	SET POLYLINE INDEX
gset_line_rep	SET POLYLINE REPRESENTATION
gset_linetype	SET LINETYPE
gset_linewidth	SET LINEWIDTH SCALE FACTOR
gset_loc_mode	SET LOCATOR MODE
gset_marker_colr_ind	SET POLYMARKER COLOUR INDEX
gset_marker_ind	SET POLYMARKER INDEX
gset_marker_rep	SET POLYMARKER REPRESENTATION
gset_marker_size	SET MARKER SIZE SCALE FACTOR
gset_marker_type	SET MARKER TYPE
gset_pat_ref_point	SET PATTERN REFERENCE POINT
gset_pat_ref_point_vecs	SET PATTERN REFERENCE POINT AND VECTORS
gset_pat_rep	SET PATTERN REPRESENTATION
gset_pat_size	SET PATTERN SIZE

Tables

Function Names

gset_pick_id	SET PICK IDENTIFIER
gset_pick_mode	SET PICK MODE
gset_seg_pri	SET SEGMENT PRIORITY
gset_seg_tran	SET SEGMENT TRANSFORMATION
gset_seg_tran3	SET SEGMENT TRANSFORMATION 3
gset_string_mode	SET STRING MODE
gset_stroke_mode	SET STROKE MODE
gset_text_align	SET TEXT ALIGNMENT
gset_text_colr_ind	SET TEXT COLOUR INDEX
gset_text_font_prec	SET TEXT FONT AND PRECISION
gset_text_ind	SET TEXT INDEX
gset_text_path	SET TEXT PATH
gset_text_rep	SET TEXT REPRESENTATION
gset_val_mode	SET VALUATOR MODE
gset_view_ind	SET VIEW INDEX
gset_view_rep3	SET VIEW REPRESENTATION 3
gset_view_tran_in_pri	SET VIEW TRANSFORMATION INPUT PRIORITY
gset_vis	SET VISIBILITY
gset_vp	SET VIEWPORT
gset_vp3	SET VIEWPORT 3
gset_vp_in_pri	SET VIEWPORT INPUT PRIORITY
gset_win	SET WINDOW
gset_win3	SET WINDOW 3
gset_ws_vp	SET WORKSTATION VIEWPORT
gset_ws_vp3	SET WORKSTATION VIEWPORT 3
gset_ws_win	SET WORKSTATION WINDOW
gset_ws_win3	SET WORKSTATION WINDOW 3
gtext	TEXT
gtext3	TEXT 3
gupd_ws	UPDATE WORKSTATION
gwrite_item	WRITE ITEM TO GKSM

4.3.2 List Ordered Alphabetically by GKS-3D Name

ACCUMULATE TRANSFORMATION MATRIX	gaccum_tran_matrix
ACCUMULATE TRANSFORMATION MATRIX 3	gaccum_tran_matrix3
ACTIVATE WORKSTATION	gactivate_ws
ASSOCIATE SEGMENT WITH WORKSTATION	gassoc_seg_ws
AWAIT EVENT	gawait_event
CELL ARRAY	gcell_array
CELL ARRAY 3	gcell_array3
CLEAR WORKSTATION	gclear_ws
CLOSE GKS	gclose_gks
CLOSE SEGMENT	gclose_seg
CLOSE WORKSTATION	gclose_ws
COPY SEGMENT FROM WORKSTATION	gcopy_seg_ws
CREATE SEGMENT	gcreate_seg
CREATE STORE (GKS-3D/C only)	gcreate_store
DEACTIVATE WORKSTATION	gdeactivate_ws
DELETE SEGMENT	gdel_seg
DELETE SEGMENT FROM WORKSTATION	gdel_seg_ws
DELETE STORE (GKS-3D/C only)	gdel_store

Function Names

Tables

EMERGENCY CLOSE GKS	gemergency_close_gks
ERROR HANDLING	gerr_hand
ERROR LOGGING	gerr_log
ESCAPE	gescape
EVALUATE TRANSFORMATION MATRIX	geval_tran_matrix
EVALUATE TRANSFORMATION MATRIX 3	geval_tran_matrix3
EVALUATE VIEW MAPPING MATRIX 3	geval_view_map_matrix3
EVALUATE VIEW ORIENTATION MATRIX 3	geval_view_ori_matrix3
FILL AREA	gfill_area
FILL AREA 3	gfill_area3
FILL AREA SET	gfill_area_set
FILL AREA SET 3	gfill_area_set3
FLUSH DEVICE EVENTS	gflush_events
GENERALIZED DRAWING PRIMITIVE	ggdp
GENERALIZED DRAWING PRIMITIVE 3	ggdp3
GET CHOICE	gget_choice
GET ITEM TYPE FROM GKSM	gget_item_type
GET LOCATOR	gget_loc
GET LOCATOR 3	gget_loc3
GET PICK	gget_pick
GET STRING	gget_string
GET STROKE	gget_stroke
GET STROKE 3	gget_stroke3
GET VALUATOR	gget_val
INITIALIZE CHOICE	ginit_choice
INITIALIZE CHOICE 3	ginit_choice3
INITIALIZE LOCATOR	ginit_loc
INITIALIZE LOCATOR 3	ginit_loc3
INITIALIZE PICK	ginit_pick
INITIALIZE PICK 3	ginit_pick3
INITIALIZE STRING	ginit_string
INITIALIZE STRING 3	ginit_string3
INITIALIZE STROKE	ginit_stroke
INITIALIZE STROKE 3	ginit_stroke3
INITIALIZE VALUATOR	ginit_val
INITIALIZE VALUATOR 3	ginit_val3
INQUIRE ASPECT SOURCE FLAGS	ginq_asfs
INQUIRE ASPECT SOURCE FLAGS 3	ginq_asfs3
INQUIRE CHARACTER BASE VECTOR	ginq_char_base_vec
INQUIRE CHARACTER EXPANSION FACTOR	ginq_char_expan
INQUIRE CHARACTER HEIGHT	ginq_char_ht
INQUIRE CHARACTER SPACING	ginq_char_space
INQUIRE CHARACTER UP VECTOR	ginq_char_up_vec
INQUIRE CHARACTER WIDTH	ginq_char_width
INQUIRE CHOICE DEVICE STATE	ginq_choice_st
INQUIRE CHOICE DEVICE STATE 3	ginq_choice_st3
INQUIRE CLIPPING	ginq_clip
INQUIRE CLIPPING 3	ginq_clip3
INQUIRE COLOUR FACILITIES	ginq_colr_fac
INQUIRE COLOUR MODEL	ginq_colr_model
INQUIRE COLOUR MODEL FACILITIES	ginq_colr_model_fac
INQUIRE COLOUR REPRESENTATION	ginq_colr_rep

Tables

Function Names

INQUIRE CURRENT HLHSR IDENTIFIER VALUE	ginq_cur_hlhrs_id
INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES	ginq_cur_indiv_attrs
INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES 3	ginq_cur_indiv_attrs3
INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER	ginq_cur_norm_tran_num
INQUIRE CURRENT PICK IDENTIFIER VALUE	ginq_cur_pick_id
INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES	ginq_cur_prim_attrs
INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES 3	ginq_cur_prim_attrs3
INQUIRE DEFAULT CHOICE DEVICE DATA	ginq_def_choice_data
INQUIRE DEFAULT CHOICE DEVICE DATA 3	ginq_def_choice_data3
INQUIRE DEFAULT DEFERRAL STATE VALUES	ginq_def_defer_sts
INQUIRE DEFAULT LOCATOR DEVICE DATA	ginq_def_loc_data
INQUIRE DEFAULT LOCATOR DEVICE DATA 3	ginq_def_loc_data3
INQUIRE DEFAULT PICK DEVICE DATA	ginq_def_pick_data
INQUIRE DEFAULT PICK DEVICE DATA 3	ginq_def_pick_data3
INQUIRE DEFAULT STRING DEVICE DATA	ginq_def_string_data
INQUIRE DEFAULT STRING DEVICE DATA 3	ginq_def_string_data3
INQUIRE DEFAULT STROKE DEVICE DATA	ginq_def_stroke_data
INQUIRE DEFAULT STROKE DEVICE DATA 3	ginq_def_stroke_data3
INQUIRE DEFAULT VALUATOR DEVICE DATA	ginq_def_val_data
INQUIRE DEFAULT VALUATOR DEVICE DATA 3	ginq_def_val_data3
INQUIRE DISPLAY SPACE SIZE	ginq_disp_space_size
INQUIRE DISPLAY SPACE SIZE 3	ginq_disp_space_size3
INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES	ginq_dyn_mod_seg_attrs
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES	ginq_dyn_mod_ws_attrs
INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3	ginq_dyn_mod_ws_attrs3
INQUIRE EDGE COLOUR INDEX	ginq_edge_colr_ind
INQUIRE EDGE FACILITIES	ginq_edge_fac
INQUIRE EDGE FLAG	ginq_edge_flag
INQUIRE EDGE INDEX	ginq_edge_ind
INQUIRE EDGE REPRESENTATION	ginq_edge_rep
INQUIRE EDGETYPE	ginq_edgetype
INQUIRE EDGEWIDTH SCALE FACTOR	ginq_edgewidth
INQUIRE FILL AREA COLOUR INDEX	ginq_fill_colr_ind
INQUIRE FILL AREA FACILITIES	ginq_fill_fac
INQUIRE FILL AREA INDEX	ginq_fill_ind
INQUIRE FILL AREA INTERIOR STYLE	ginq_fill_int_style
INQUIRE FILL AREA REPRESENTATION	ginq_fill_rep
INQUIRE FILL AREA STYLE INDEX	ginq_fill_style_ind
INQUIRE GENERALIZED DRAWING PRIMITIVE	ginq_gdp
INQUIRE GENERALIZED DRAWING PRIMITIVE 3	ginq_gdp3
INQUIRE HLHSR IDENTIFIER FACILITIES	ginq_hlhrs_id_fac
INQUIRE HLHSR MODE	ginq_hlhrs_mode
INQUIRE HLHSR MODE FACILITIES	ginq_hlhrs_mode_fac
INQUIRE INPUT QUEUE OVERFLOW	ginq_in_overf
INQUIRE LEVEL OF GKS	ginq_level_gks
INQUIRE LINETYPE	ginq_linetype
INQUIRE LINEWIDTH SCALE FACTOR	ginq_linewidth
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES	ginq_list_avail_gdps
INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3	ginq_list_avail_gdps3
INQUIRE LIST OF AVAILABLE WORKSTATION TYPES	ginq_list_avail_ws_types
INQUIRE LIST OF COLOUR INDICES	ginq_list_colr_inds
INQUIRE LIST OF EDGE INDICES	ginq_list_edge_inds

Function Names

Tables

INQUIRE LIST OF FILL AREA INDICES	ginq_list_fill_inds
INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS	ginq_list_norm_tran_nums
INQUIRE LIST OF PATTERN INDICES	ginq_list_pat_inds
INQUIRE LIST OF POLYLINE INDICES	ginq_list_line_inds
INQUIRE LIST OF POLYMARKER INDICES	ginq_list_marker_inds
INQUIRE LIST OF TEXT INDICES	ginq_list_text_inds
INQUIRE LIST OF VIEW INDICES	ginq_list_view_inds
INQUIRE LOCATOR DEVICE STATE	ginq_loc_st
INQUIRE LOCATOR DEVICE STATE 3	ginq_loc_st3
INQUIRE MARKER SIZE SCALE FACTOR	ginq_marker_size
INQUIRE MARKER TYPE	ginq_marker_type
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES	ginq_max_ws_st_tables
INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3	ginq_max_ws_st_tables3
INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER	ginq_max_norm_tran_num
INQUIRE MORE SIMULTANEOUS EVENTS	ginq_more_simult_events
INQUIRE NAME OF OPEN SEGMENT	ginq_name_open_seg
INQUIRE NORMALIZATION TRANSFORMATION	ginq_norm_tran
INQUIRE NORMALIZATION TRANSFORMATION 3	ginq_norm_tran3
INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES	ginq_num_avail_in
INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED	ginq_num_seg_pris
INQUIRE OPERATING STATE VALUE	ginq_op_st
INQUIRE PATTERN FACILITIES	ginq_pat_fac
INQUIRE PATTERN HEIGHT VECTOR	ginq_pat_ht_vec
INQUIRE PATTERN REFERENCE POINT	ginq_pat_ref_point
INQUIRE PATTERN REFERENCE POINT AND VECTORS	ginq_pat_ref_point_vecs
INQUIRE PATTERN REPRESENTATION	ginq_pat_rep
INQUIRE PATTERN WIDTH VECTOR	ginq_pat_width_vec
INQUIRE PICK DEVICE STATE	ginq_pick_st
INQUIRE PICK DEVICE STATE 3	ginq_pick_st3
INQUIRE PIXEL	ginq_pixel
INQUIRE PIXEL ARRAY	ginq_pixel_array
INQUIRE PIXEL ARRAY DIMENSIONS	ginq_pixel_array_dims
INQUIRE POLYLINE COLOUR INDEX	ginq_line_colr_ind
INQUIRE POLYLINE FACILITIES	ginq_line_fac
INQUIRE POLYLINE INDEX	ginq_line_ind
INQUIRE POLYLINE REPRESENTATION	ginq_line_rep
INQUIRE POLYMARKER COLOUR INDEX	ginq_marker_colr_ind
INQUIRE POLYMARKER FACILITIES	ginq_marker_fac
INQUIRE POLYMARKER INDEX	ginq_marker_ind
INQUIRE POLYMARKER REPRESENTATION	ginq_marker_rep
INQUIRE PREDEFINED COLOUR REPRESENTATION	ginq_pred_colr_rep
INQUIRE PREDEFINED EDGE REPRESENTATION	ginq_pred_edge_rep
INQUIRE PREDEFINED FILL AREA REPRESENTATION	ginq_pred_fill_rep
INQUIRE PREDEFINED PATTERN REPRESENTATION	ginq_pred_pat_rep
INQUIRE PREDEFINED POLYLINE REPRESENTATION	ginq_pred_line_rep
INQUIRE PREDEFINED POLYMARKER REPRESENTATION	ginq_pred_marker_rep
INQUIRE PREDEFINED TEXT REPRESENTATION	ginq_pred_text_rep
INQUIRE PREDEFINED VIEW REPRESENTATION	ginq_pred_view_rep
INQUIRE SEGMENT ATTRIBUTES	ginq_seg_attr
INQUIRE SEGMENT ATTRIBUTES 3	ginq_seg_attr3
INQUIRE SET OF ACTIVE WORKSTATIONS	ginq_set_active_wss
INQUIRE SET OF ASSOCIATED WORKSTATIONS	ginq_set_assoc_wss

Tables

Function Names

INQUIRE SET OF OPEN WORKSTATIONS	ginq_set_open_wss
INQUIRE SET OF SEGMENT NAMES IN USE	ginq_set_seg_names
INQUIRE SET OF SEGMENT NAMES ON WORKSTATION	ginq_set_seg_names_ws
INQUIRE STRING DEVICE STATE	ginq_string_st
INQUIRE STRING DEVICE STATE 3	ginq_string_st3
INQUIRE STROKE DEVICE STATE	ginq_stroke_st
INQUIRE STROKE DEVICE STATE 3	ginq_stroke_st3
INQUIRE TEXT ALIGNMENT	ginq_text_align
INQUIRE TEXT COLOUR INDEX	ginq_text_colr_ind
INQUIRE TEXT EXTENT	ginq_text_extent
INQUIRE TEXT EXTENT 3	ginq_text_extent3
INQUIRE TEXT FACILITIES	ginq_text_fac
INQUIRE TEXT FONT AND PRECISION	ginq_text_font_prec
INQUIRE TEXT INDEX	ginq_text_ind
INQUIRE TEXT PATH	ginq_text_path
INQUIRE TEXT REPRESENTATION	ginq_text_rep
INQUIRE VALUATOR DEVICE STATE	ginq_val_st
INQUIRE VALUATOR DEVICE STATE 3	ginq_val_st3
INQUIRE VIEW FACILITIES	ginq_view_fac
INQUIRE VIEW INDEX	ginq_view_ind
INQUIRE VIEW REPRESENTATION 3	ginq_view_rep3
INQUIRE WORKSTATION CATEGORY	ginq_ws_cat
INQUIRE WORKSTATION CLASSIFICATION	ginq_ws_class
INQUIRE WORKSTATION CONNECTION AND TYPE	ginq_ws_conn_type
INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES	ginq_ws_defer_upd_sts
INQUIRE WORKSTATION MAXIMUM NUMBERS	ginq_ws_max_nums
INQUIRE WORKSTATION STATE	ginq_ws_st
INQUIRE WORKSTATION TRANSFORMATION	ginq_ws_tran
INQUIRE WORKSTATION TRANSFORMATION 3	ginq_ws_tran3
INSERT SEGMENT	ginsert_seg
INSERT SEGMENT 3	ginsert_seg3
INTERPRET ITEM	ginterpret_item
MESSAGE	gmessage
OPEN GKS	gopen_gks
OPEN WORKSTATION	gopen_ws
POLYLINE	gpolyline
POLYLINE 3	gpolyline3
POLYMARKER	gpolymarker
POLYMARKER 3	gpolymarker3
READ ITEM FROM GKSM	gread_item
REDRAW ALL SEGMENTS ON WORKSTATION	gredraw_all_segs_ws
RENAME SEGMENT	grename_seg
REQUEST CHOICE	greg_choice
REQUEST LOCATOR	greg_loc
REQUEST LOCATOR 3	greg_loc3
REQUEST PICK	greg_pick
REQUEST STRING	greg_string
REQUEST STROKE	greg_stroke
REQUEST STROKE 3	greg_stroke3
REQUEST VALUATOR	greg_val
SAMPLE CHOICE	gsample_choice
SAMPLE LOCATOR	gsample_loc

Function Names

Tables

SAMPLE LOCATOR 3	gsample_loc3
SAMPLE PICK	gsample_pick
SAMPLE STRING	gsample_string
SAMPLE STROKE	gsample_stroke
SAMPLE STROKE 3	gsample_stroke3
SAMPLE VALUATOR	gsample_val
SELECT NORMALIZATION TRANSFORMATION	gsel_norm_tran
SET ASPECT SOURCE FLAGS	gset_asfs
SET ASPECT SOURCE FLAGS	gset_asfs3
SET CHARACTER EXPANSION FACTOR	gset_char_expan
SET CHARACTER HEIGHT	gset_char_ht
SET CHARACTER SPACING	gset_char_space
SET CHARACTER UP VECTOR	gset_char_up_vec
SET CHOICE MODE	gset_choice_mode
SET CLIPPING INDICATOR	gset_clip_ind
SET COLOUR MODEL	gset_colr_model
SET COLOUR REPRESENTATION	gset_colr_rep
SET DEFERRAL STATE	gset_defer_st
SET DETECTABILITY	gset_det
SET EDGE COLOUR INDEX	gset_edge_colr_ind
SET EDGE FLAG	gset_edge_flag
SET EDGE INDEX	gset_edge_ind
SET EDGE REPRESENTATION	gset_edge_rep
SET EDGETYPE	gset_edgetype
SET EDGEWIDTH SCALE FACTOR	gset_edgewidth
SET ERROR HANDLER (GKS-3D/C only)	gset_err_hand
SET FILL AREA COLOUR INDEX	gset_fill_colr_ind
SET FILL AREA INDEX	gset_fill_ind
SET FILL AREA INTERIOR STYLE	gset_fill_int_style
SET FILL AREA REPRESENTATION	gset_fill_rep
SET FILL AREA STYLE INDEX	gset_fill_style_ind
SET HIGHLIGHTING	gset_highl
SET HLHSR IDENTIFIER	gset_hlhsr_id
SET HLHSR MODE	gset_hlhsr_mode
SET LINETYPE	gset_linetype
SET LINEWIDTH SCALE FACTOR	gset_linewidth
SET LOCATOR MODE	gset_loc_mode
SET MARKER SIZE SCALE FACTOR	gset_marker_size
SET MARKER TYPE	gset_marker_type
SET PATTERN REFERENCE POINT	gset_pat_ref_point
SET PATTERN REFERENCE POINT AND VECTORS	gset_pat_ref_point_vecs
SET PATTERN REPRESENTATION	gset_pat_rep
SET PATTERN SIZE	gset_pat_size
SET PICK IDENTIFIER	gset_pick_id
SET PICK MODE	gset_pick_mode
SET POLYLINE COLOUR INDEX	gset_line_colr_ind
SET POLYLINE INDEX	gset_line_ind
SET POLYLINE REPRESENTATION	gset_line_rep
SET POLYMARKER COLOUR INDEX	gset_marker_colr_ind
SET POLYMARKER INDEX	gset_marker_ind
SET POLYMARKER REPRESENTATION	gset_marker_rep
SET SEGMENT PRIORITY	gset_seg_pri

Tables**Function Names**

SET SEGMENT TRANSFORMATION	gset_seg_tran
SET SEGMENT TRANSFORMATION 3	gset_seg_tran3
SET STRING MODE	gset_string_mode
SET STROKE MODE	gset_stroke_mode
SET TEXT ALIGNMENT	gset_text_align
SET TEXT COLOUR INDEX	gset_text_colr_ind
SET TEXT FONT AND PRECISION	gset_text_font_prec
SET TEXT INDEX	gset_text_ind
SET TEXT PATH	gset_text_path
SET TEXT REPRESENTATION	gset_text_rep
SET VALUATOR MODE	gset_val_mode
SET VIEW INDEX	gset_view_ind
SET VIEW REPRESENTATION 3	gset_view_rep3
SET VIEW TRANSFORMATION INPUT PRIORITY	gset_view_tran_in_pri
SET VIEWPORT	gset_vp
SET VIEWPORT 3	gset_vp3
SET VIEWPORT INPUT PRIORITY	gset_vp_in_pri
SET VISIBILITY	gset_vis
SET WINDOW	gset_win
SET WINDOW 3	gset_win3
SET WORKSTATION VIEWPORT	gset_ws_vp
SET WORKSTATION VIEWPORT 3	gset_ws_vp3
SET WORKSTATION WINDOW	gset_ws_win
SET WORKSTATION WINDOW 3	gset_ws_win3
TEXT	gtext
TEXT 3	gtext3
UPDATE WORKSTATION	gupd_ws
WRITE ITEM TO GKSM	gwrite_item

4.3.3 List Ordered Alphabetically by Bound Name within Level**4.3.3.1 Level 0A**

gactivate_ws	ACTIVATE WORKSTATION
gcell_array	CELL ARRAY
gcell_array3	CELL ARRAY 3
gclear_ws	CLEAR WORKSTATION
gclose_gks	CLOSE GKS
gclose_ws	CLOSE WORKSTATION
gcreate_store	CREATE STORE (GKS-3D/C only)
gdeactivate_ws	DEACTIVATE WORKSTATION
gdel_store	DELETE STORE (GKS-3D/C only)
gemergency_close_gks	EMERGENCY CLOSE GKS
gerr_hand	ERROR HANDLING
gerr_log	ERROR LOGGING
gescape	ESCAPE
geval_view_map_matrix3	EVALUATE VIEW MAPPING MATRIX 3
geval_view_ori_matrix3	EVALUATE VIEW ORIENTATION MATRIX 3
gfill_area	FILL AREA
gfill_area3	FILL AREA 3
gfill_area_set	FILL AREA SET
gfill_area_set3	FILL AREA SET 3

Function Names

Tables

ggdp	GENERALIZED DRAWING PRIMITIVE
ggdp3	GENERALIZED DRAWING PRIMITIVE 3
gget_item_type	GET ITEM TYPE FROM GKSM
ginq_asfs	INQUIRE ASPECT SOURCE FLAGS
ginq_asfs3	INQUIRE ASPECT SOURCE FLAGS 3
ginq_char_base_vec	INQUIRE CHARACTER BASE VECTOR
ginq_char_expan	INQUIRE CHARACTER EXPANSION FACTOR
ginq_char_ht	INQUIRE CHARACTER HEIGHT
ginq_char_space	INQUIRE CHARACTER SPACING
ginq_char_up_vec	INQUIRE CHARACTER UP VECTOR
ginq_char_width	INQUIRE CHARACTER WIDTH
ginq_clip	INQUIRE CLIPPING
ginq_clip3	INQUIRE CLIPPING 3
ginq_colr_fac	INQUIRE COLOUR FACILITIES
ginq_colr_model	INQUIRE COLOUR MODEL
ginq_colr_model_fac	INQUIRE COLOUR MODEL FACILITIES
ginq_colr_rep	INQUIRE COLOUR REPRESENTATION
ginq_cur_indiv_attr	INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES
ginq_cur_indiv_attr3	INQUIRE CURRENT INDIVIDUAL ATTRIBUTE VALUES 3
ginq_cur_norm_tran_num	INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER
ginq_cur_prim_attr	INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES
ginq_cur_prim_attr3	INQUIRE CURRENT PRIMITIVE ATTRIBUTE VALUES 3
ginq_disp_space_size	INQUIRE DISPLAY SPACE SIZE
ginq_disp_space_size3	INQUIRE DISPLAY SPACE SIZE 3
ginq_dyn_mod_ws_attr	INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES
ginq_dyn_mod_ws_attr3	INQUIRE DYNAMIC MODIFICATION OF WORKSTATION ATTRIBUTES 3
ginq_edge_colr_ind	INQUIRE EDGE COLOUR INDEX
ginq_edge_fac	INQUIRE EDGE FACILITIES
ginq_edge_flag	INQUIRE EDGE FLAG
ginq_edge_ind	INQUIRE EDGE INDEX
ginq_edgetype	INQUIRE EDGETYPE
ginq_edgewidth	INQUIRE EDGEWIDTH SCALE FACTOR
ginq_fill_colr_ind	INQUIRE FILL AREA COLOUR INDEX
ginq_fill_fac	INQUIRE FILL AREA FACILITIES
ginq_fill_ind	INQUIRE FILL AREA INDEX
ginq_fill_int_style	INQUIRE FILL AREA INTERIOR STYLE
ginq_fill_style_ind	INQUIRE FILL AREA STYLE INDEX
ginq_gdp	INQUIRE GENERALIZED DRAWING PRIMITIVE
ginq_gdp3	INQUIRE GENERALIZED DRAWING PRIMITIVE 3
ginq_level_gks	INQUIRE LEVEL OF GKS
ginq_linetype	INQUIRE LINETYPE
ginq_linewidth	INQUIRE LINewidth SCALE FACTOR
ginq_list_avail_gdps	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES
ginq_list_avail_gdps3	INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES 3
ginq_list_avail_ws_types	INQUIRE LIST OF AVAILABLE WORKSTATION TYPES
ginq_list_colr_inds	INQUIRE LIST OF COLOUR INDICES
ginq_list_norm_tran_nums	INQUIRE LIST OF NORMALIZATION TRANSFORMATION NUMBERS
ginq_list_view_inds	INQUIRE LIST OF VIEW INDICES
ginq_marker_size	INQUIRE MARKER SIZE SCALE FACTOR
ginq_marker_type	INQUIRE MARKER TYPE
ginq_max_ws_st_tables	INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES
ginq_max_ws_st_tables3	INQUIRE MAXIMUM LENGTH OF WORKSTATION STATE TABLES 3

Tables

Function Names

ginq_max_norm_tran_num	INQUIRE MAXIMUM NORMALIZATION TRANSFORMATION NUMBER
ginq_norm_tran	INQUIRE NORMALIZATION TRANSFORMATION
ginq_norm_tran3	INQUIRE NORMALIZATION TRANSFORMATION 3
ginq_op_st	INQUIRE OPERATING STATE VALUE
ginq_pat_fac	INQUIRE PATTERN FACILITIES
ginq_pat_ht_vec	INQUIRE PATTERN HEIGHT VECTOR
ginq_pat_ref_point	INQUIRE PATTERN REFERENCE POINT
ginq_pat_ref_point_vecs	INQUIRE PATTERN REFERENCE POINT AND VECTORS
ginq_pat_width_vec	INQUIRE PATTERN WIDTH VECTOR
ginq_pixel	INQUIRE PIXEL
ginq_pixel_array	INQUIRE PIXEL ARRAY
ginq_pixel_array_dims	INQUIRE PIXEL ARRAY DIMENSIONS
ginq_line_colr_ind	INQUIRE POLYLINE COLOUR INDEX
ginq_line_fac	INQUIRE POLYLINE FACILITIES
ginq_line_ind	INQUIRE POLYLINE INDEX
ginq_marker_colr_ind	INQUIRE POLYMARKER COLOUR INDEX
ginq_marker_fac	INQUIRE POLYMARKER FACILITIES
ginq_marker_ind	INQUIRE POLYMARKER INDEX
ginq_pred_colr_rep	INQUIRE PREDEFINED COLOUR REPRESENTATION
ginq_pred_edge_rep	INQUIRE PREDEFINED EDGE REPRESENTATION
ginq_pred_fill_rep	INQUIRE PREDEFINED FILL AREA REPRESENTATION
ginq_pred_pat_rep	INQUIRE PREDEFINED PATTERN REPRESENTATION
ginq_pred_line_rep	INQUIRE PREDEFINED POLYLINE REPRESENTATION
ginq_pred_marker_rep	INQUIRE PREDEFINED POLYMARKER REPRESENTATION
ginq_pred_text_rep	INQUIRE PREDEFINED TEXT REPRESENTATION
ginq_pred_view_rep	INQUIRE PREDEFINED VIEW REPRESENTATION
ginq_set_active_wss	INQUIRE SET OF ACTIVE WORKSTATIONS
ginq_set_open_wss	INQUIRE SET OF OPEN WORKSTATIONS
ginq_text_align	INQUIRE TEXT ALIGNMENT
ginq_text_colr_ind	INQUIRE TEXT COLOUR INDEX
ginq_text_extent	INQUIRE TEXT EXTENT
ginq_text_extent3	INQUIRE TEXT EXTENT 3
ginq_text_fac	INQUIRE TEXT FACILITIES
ginq_text_font_prec	INQUIRE TEXT FONT AND PRECISION
ginq_text_ind	INQUIRE TEXT INDEX
ginq_text_path	INQUIRE TEXT PATH
ginq_view_fac	INQUIRE VIEW FACILITIES
ginq_view_ind	INQUIRE VIEW INDEX
ginq_view_rep3	INQUIRE VIEW REPRESENTATION 3
ginq_ws_cat	INQUIRE WORKSTATION CATEGORY
ginq_ws_class	INQUIRE WORKSTATION CLASSIFICATION
ginq_ws_conn_type	INQUIRE WORKSTATION CONNECTION AND TYPE
ginq_ws_defer_upd_sts	INQUIRE WORKSTATION DEFERRAL AND UPDATE STATES
ginq_ws_max_nums	INQUIRE WORKSTATION MAXIMUM NUMBERS
ginq_ws_st	INQUIRE WORKSTATION STATE
ginq_ws_tran	INQUIRE WORKSTATION TRANSFORMATION
ginq_ws_tran3	INQUIRE WORKSTATION TRANSFORMATION 3
ginterpret_item	INTERPRET ITEM
gopen_gks	OPEN GKS
gopen_ws	OPEN WORKSTATION
gpolyline	POLYLINE
gpolyline3	POLYLINE 3

Function Names

Tables

gpolymarker	POLYMARKER
gpolymarker3	POLYMARKER 3
gread_item	READ ITEM FROM GKSM
gset_norm_tran	SELECT NORMALIZATION TRANSFORMATION
gset_asfs	SET ASPECT SOURCE FLAGS
gset_asfs3	SET ASPECT SOURCE FLAGS
gset_char_expan	SET CHARACTER EXPANSION FACTOR
gset_char_ht	SET CHARACTER HEIGHT
gset_char_space	SET CHARACTER SPACING
gset_char_up_vec	SET CHARACTER UP VECTOR
gset_clip_ind	SET CLIPPING INDICATOR
gset_colr_model	SET COLOUR MODEL
gset_colr_rep	SET COLOUR REPRESENTATION
gset_edge_colr_ind	SET EDGE COLOUR INDEX
gset_edge_flag	SET EDGE FLAG
gset_edge_ind	SET EDGE INDEX
gset_edgetype	SET EDGETYPE
gset_edgewidth	SET EDGEWIDTH SCALE FACTOR
gset_err_hand	SET ERROR HANDLER (GKS-3D/C only)
gset_fill_colr_ind	SET FILL AREA COLOUR INDEX
gset_fill_ind	SET FILL AREA INDEX
gset_fill_int_style	SET FILL AREA INTERIOR STYLE
gset_fill_style_ind	SET FILL AREA STYLE INDEX
gset_linetype	SET LINETYPE
gset_linewidth	SET LINEWIDTH SCALE FACTOR
gset_marker_size	SET MARKER SIZE SCALE FACTOR
gset_marker_type	SET MARKER TYPE
gset_pat_ref_point	SET PATTERN REFERENCE POINT
gset_pat_ref_point_vecs	SET PATTERN REFERENCE POINT AND VECTORS
gset_pat_size	SET PATTERN SIZE
gset_line_colr_ind	SET POLYLINE COLOUR INDEX
gset_line_ind	SET POLYLINE INDEX
gset_marker_colr_ind	SET POLYMARKER COLOUR INDEX
gset_marker_ind	SET POLYMARKER INDEX
gset_text_align	SET TEXT ALIGNMENT
gset_text_colr_ind	SET TEXT COLOUR INDEX
gset_text_font_prec	SET TEXT FONT AND PRECISION
gset_text_ind	SET TEXT INDEX
gset_text_path	SET TEXT PATH
gset_view_ind	SET VIEW INDEX
gset_view_rep3	SET VIEW REPRESENTATION 3
gset_vp	SET VIEWPORT
gset_vp3	SET VIEWPORT 3
gset_win	SET WINDOW
gset_win3	SET WINDOW 3
gset_ws_vp	SET WORKSTATION VIEWPORT
gset_ws_vp3	SET WORKSTATION VIEWPORT 3
gset_ws_win	SET WORKSTATION WINDOW
gset_ws_win3	SET WORKSTATION WINDOW 3
gtext	TEXT
gtext3	TEXT 3
gupd_ws	UPDATE WORKSTATION

Tables

Function Names

gwrite_item

WRITE ITEM TO GKSM

4.3.3.2 Level 0B

ginit_choice	INITIALIZE CHOICE
ginit_choice3	INITIALIZE CHOICE 3
ginit_loc	INITIALIZE LOCATOR
ginit_loc3	INITIALIZE LOCATOR 3
ginit_string	INITIALIZE STRING
ginit_string3	INITIALIZE STRING 3
ginit_stroke	INITIALIZE STROKE
ginit_stroke3	INITIALIZE STROKE 3
ginit_val	INITIALIZE VALUATOR
ginit_val3	INITIALIZE VALUATOR 3
ginq_choice_st	INQUIRE CHOICE DEVICE STATE
ginq_choice_st3	INQUIRE CHOICE DEVICE STATE 3
ginq_def_choice_data	INQUIRE DEFAULT CHOICE DEVICE DATA
ginq_def_choice_data3	INQUIRE DEFAULT CHOICE DEVICE DATA 3
ginq_def_loc_data	INQUIRE DEFAULT LOCATOR DEVICE DATA
ginq_def_loc_data3	INQUIRE DEFAULT LOCATOR DEVICE DATA 3
ginq_def_string_data	INQUIRE DEFAULT STRING DEVICE DATA
ginq_def_string_data3	INQUIRE DEFAULT STRING DEVICE DATA 3
ginq_def_stroke_data	INQUIRE DEFAULT STROKE DEVICE DATA
ginq_def_stroke_data3	INQUIRE DEFAULT STROKE DEVICE DATA 3
ginq_def_val_data	INQUIRE DEFAULT VALUATOR DEVICE DATA
ginq_def_val_data3	INQUIRE DEFAULT VALUATOR DEVICE DATA 3
ginq_loc_st	INQUIRE LOCATOR DEVICE STATE
ginq_loc_st3	INQUIRE LOCATOR DEVICE STATE 3
ginq_num_avail_in	INQUIRE NUMBER OF AVAILABLE LOGICAL INPUT DEVICES
ginq_string_st	INQUIRE STRING DEVICE STATE
ginq_string_st3	INQUIRE STRING DEVICE STATE 3
ginq_stroke_st	INQUIRE STROKE DEVICE STATE
ginq_stroke_st3	INQUIRE STROKE DEVICE STATE 3
ginq_val_st	INQUIRE VALUATOR DEVICE STATE
ginq_val_st3	INQUIRE VALUATOR DEVICE STATE 3
greq_choice	REQUEST CHOICE
greq_loc	REQUEST LOCATOR
greq_loc3	REQUEST LOCATOR 3
greq_string	REQUEST STRING
greq_stroke	REQUEST STROKE
greq_stroke3	REQUEST STROKE 3
greq_val	REQUEST VALUATOR
gset_choice_mode	SET CHOICE MODE
gset_loc_mode	SET LOCATOR MODE
gset_string_mode	SET STRING MODE
gset_stroke_mode	SET STROKE MODE
gset_val_mode	SET VALUATOR MODE
gset_view_tran_in_pri	SET VIEW TRANSFORMATION INPUT PRIORITY
gset_vp_in_pri	SET VIEWPORT INPUT PRIORITY

Function Names

Tables

4.3.3.3 Level 0C

gawait_event	AWAIT EVENT
gflush_events	FLUSH DEVICE EVENTS
gget_choice	GET CHOICE
gget_loc	GET LOCATOR
gget_loc3	GET LOCATOR 3
gget_string	GET STRING
gget_stroke	GET STROKE
gget_stroke3	GET STROKE 3
gget_val	GET VALUATOR
ginq_in_overf	INQUIRE INPUT QUEUE OVERFLOW
ginq_more_simult_events	INQUIRE MORE SIMULTANEOUS EVENTS
gsample_choice	SAMPLE CHOICE
gsample_loc	SAMPLE LOCATOR
gsample_loc3	SAMPLE LOCATOR 3
gsample_string	SAMPLE STRING
gsample_stroke	SAMPLE STROKE
gsample_stroke3	SAMPLE STROKE 3
gsample_val	SAMPLE VALUATOR

4.3.3.4 Level 1A

gaccum_tran_matrix	ACCUMULATE TRANSFORMATION MATRIX
gaccum_tran_matrix3	ACCUMULATE TRANSFORMATION MATRIX 3
gclose_seg	CLOSE SEGMENT
gcreate_seg	CREATE SEGMENT
gdel_seg	DELETE SEGMENT
gdel_seg_ws	DELETE SEGMENT FROM WORKSTATION
geval_tran_matrix	EVALUATE TRANSFORMATION MATRIX
geval_tran_matrix3	EVALUATE TRANSFORMATION MATRIX 3
ginq_cur_hlhrs_id	INQUIRE CURRENT HLHSR IDENTIFIER VALUE
ginq_def_defer_sts	INQUIRE DEFAULT DEFERRAL STATE VALUES
ginq_dyn_mod_seg_attrs	INQUIRE DYNAMIC MODIFICATION OF SEGMENT ATTRIBUTES
ginq_edge_rep	INQUIRE EDGE REPRESENTATION
ginq_fill_rep	INQUIRE FILL AREA REPRESENTATION
ginq_hlhrs_id_fac	INQUIRE HLHSR IDENTIFIER FACILITIES
ginq_hlhrs_mode	INQUIRE HLHSR MODE
ginq_hlhrs_mode_fac	INQUIRE HLHSR MODE FACILITIES
ginq_list_edge_inds	INQUIRE LIST OF EDGE INDICES
ginq_list_fill_inds	INQUIRE LIST OF FILL AREA INDICES
ginq_list_pat_inds	INQUIRE LIST OF PATTERN INDICES
ginq_list_line_inds	INQUIRE LIST OF POLYLINE INDICES
ginq_list_marker_inds	INQUIRE LIST OF POLYMARKER INDICES
ginq_list_text_inds	INQUIRE LIST OF TEXT INDICES
ginq_name_open_seg	INQUIRE NAME OF OPEN SEGMENT
ginq_num_seg_pris	INQUIRE NUMBER OF SEGMENT PRIORITIES SUPPORTED
ginq_pat_rep	INQUIRE PATTERN REPRESENTATION
ginq_line_rep	INQUIRE POLYLINE REPRESENTATION
ginq_marker_rep	INQUIRE POLYMARKER REPRESENTATION
ginq_seg_attrs	INQUIRE SEGMENT ATTRIBUTES
ginq_seg_attrs3	INQUIRE SEGMENT ATTRIBUTES 3
ginq_set_assoc_wss	INQUIRE SET OF ASSOCIATED WORKSTATIONS

Tables

Function Names

ginq_set_seg_names	INQUIRE SET OF SEGMENT NAMES IN USE
ginq_set_seg_names_ws	INQUIRE SET OF SEGMENT NAMES ON WORKSTATION
ginq_text_rep	INQUIRE TEXT REPRESENTATION
gmessage	MESSAGE
gredraw_all_segs_ws	REDRAW ALL SEGMENTS ON WORKSTATION
grename_seg	RENAME SEGMENT
gset_defer_st	SET DEFERRAL STATE
gset_edge_rep	SET EDGE REPRESENTATION
gset_fill_rep	SET FILL AREA REPRESENTATION
gset_highl	SET HIGHLIGHTING
gset_hlhr_id	SET HLHR IDENTIFIER
gset_hlhr_mode	SET HLHR MODE
gset_pat_rep	SET PATTERN REPRESENTATION
gset_line_rep	SET POLYLINE REPRESENTATION
gset_marker_rep	SET POLYMARKER REPRESENTATION
gset_seg_pri	SET SEGMENT PRIORITY
gset_seg_tran	SET SEGMENT TRANSFORMATION
gset_seg_tran3	SET SEGMENT TRANSFORMATION 3
gset_text_rep	SET TEXT REPRESENTATION
gset_vis	SET VISIBILITY

4.3.3.5 Level 1B

ginit_pick	INITIALIZE PICK
ginit_pick3	INITIALIZE PICK 3
ginq_cur_pick_id	INQUIRE CURRENT PICK IDENTIFIER VALUE
ginq_def_pick_data	INQUIRE DEFAULT PICK DEVICE DATA
ginq_def_pick_data3	INQUIRE DEFAULT PICK DEVICE DATA 3
ginq_pick_st	INQUIRE PICK DEVICE STATE
ginq_pick_st3	INQUIRE PICK DEVICE STATE 3
greq_pick	REQUEST PICK
gset_det	SET DETECTABILITY
gset_pick_id	SET PICK IDENTIFIER
gset_pick_mode	SET PICK MODE

4.3.3.6 Level 1C

gget_pick	GET PICK
gsample_pick	SAMPLE PICK

4.3.3.7 Level 2A

gassoc_seg_ws	ASSOCIATE SEGMENT WITH WORKSTATION
gcopy_seg_ws	COPY SEGMENT FROM WORKSTATION
ginsert_seg	INSERT SEGMENT
ginsert_seg3	INSERT SEGMENT 3

5 Type Definitions

5.1 Mapping of GKS-3D data types

The GKS-3D document specifies a set of abstract data types. This clause gives the mapping from those data types to the data types defined in this part of ISO/IEC 8806.

GKS-3D data type		C binding data type
I	integer	Gint
R	real	Gfloat
S	string	char *
P	2D point	Gpoint
P3	3D point	Gpoint3
L	list of 2D points	Gpoint_list
L3	list of 3D points	Gpoint_list3
2*R	2D vector	Gvec, Gfloat_size
3*R	3D vector	Gvec3, Gfloat_size3
N	name	Gint, char *, void *
E	enumeration type	typedef enum
CLR	colour specification	Gcolr_rep

5.2 Environmental Type Definitions

The data types defined in this section allow for the ease of porting GKS-3D/C implementations between different environments. These types are used as the basis for all the other data types.

An implementation shall document the C types used for GKS-3D/C types Gfloat and Gint.

Gfloat floating point number

This data type shall be defined by the implementation as a floating point type suitable for use within GKS-3D/C. Suggest:

```
typedef float Gfloat;
```

Gint integer

This data type shall be defined by the implementation as a signed integral type suitable for use within GKS-3D/C. Suggest:

```
typedef int Gint;
```

size_t

An additional environmental data type is the type `size_t`. Depending on the environment, `size_t` can be defined as `int`, `unsigned`, `unsigned long`, etc. `size_t` is defined in the `#include` file `<stddef.h>`.

Type Definitions

Implementation Dependent Type Definitions

5.3 Implementation Dependent Type Definitions

This subclause presents the skeleton structure of the implementation dependent data types.

REMARKS:

- In order to produce syntactically correct data types, most implementation and registration dependent parts are of type `Gdata`. They can be replaced by any other structure.
- The data records of registered prompt and echo types are represented by structures with the names `pet_rn` (n the prompt and echo type), whereas the data records of unregistered prompt and echo types are represented by structures with the names `pet_un` (-n the prompt and echo type).
- The data records of registered and unregistered GDPs, escape input and escape output are denoted in a similar way.
- For the input data records, the mandatory part, as specified in GKS-3D (ISO 8805), is defined outside the union part of `Gxxx_data`. This mandatory part is implementation independent. Some of the `pet_rn` fields are given as structures of other type than `Gdata` for illustrative reasons. Thus, e.g. the `pet_r2` field of `Gchoice_data` contains a list of prompts, because GKS-3D defines prompt and echo type 2 this way for the CHOICE device;
- For illustrative reasons, `Gitem_data` is sketchily (and incompletely) defined such that the items described in ISO 8805, Annex E could be represented by GKS-3D/C data types. This definition is, however, not mandatory. Anyway, `Gitem_data` shall be defined in such a way that items which correspond with unsupported GDP's can be read from GKSM and written to GKSM.

Implementation Dependent Type Definitions

Type Definitions

Gchoice_data choice data record

```

typedef struct {
    union Gchoice_pets {
        struct Gchoice_pet_r1 {
            Gdata      impl_dep;          /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gchoice_pet_r2 {
            Gint        num_prs;          /* number of prompts */
            Gpr_flag    *prs;             /* prompt array */
            Gdata      impl_dep;          /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gchoice_pet_r3 {
            Gint        num_strings;      /* number of choice strings */
            char        **strings;        /* array of choice strings */
            Gdata      impl_dep;          /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gchoice_pet_r4 {
            Gint        num_strings;      /* number of choice strings */
            char        **strings;        /* array of choice strings */
            Gdata      impl_dep;          /* impl. dep. */
        } pet_r4; /* pet 4 data */
        struct Gchoice_pet_r5 {
            Gint        seg_name;         /* segment name */
            Gint        num_pick_ids;     /* number of pick identifiers */
            Gint        *pick_ids;        /* array of pick identifiers */
            Gdata      impl_dep;          /* impl. dep. */
        } pet_r5; /* pet 5 data */
        struct Gchoice_pet_ul {
            Gdata      impl_dep;          /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* . . . impl. defined PET's */
    } pets;
} Gchoice_data;

```

Type Definitions

Implementation Dependent Type Definitions

Gchoice_data3 choice data record 3

```

typedef struct {
    union Gchoice3_pets {
        struct Gchoice3_pet_r1 {
            Gdata    impl_dep;        /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gchoice3_pet_r2 {
            Gint      num_prs;        /* number of prompts */
            Gpr_flag  *prs;          /* prompt array */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gchoice3_pet_r3 {
            Gint      num_strings;    /* number of choice strings */
            char      **strings;      /* array of choice strings */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gchoice3_pet_r4 {
            Gint      num_strings;    /* number of choice strings */
            char      **strings;      /* array of choice strings */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r4; /* pet 4 data */
        struct Gchoice3_pet_r5 {
            Gint      seg_name;       /* segment name */
            Gint      num_pick_ids;   /* number of pick identifiers */
            Gint      *pick_ids;     /* array of pick identifiers */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r5; /* pet 5 data */
        struct Gchoice3_pet_ul {
            Gdata      impl_dep;      /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* . . . impl. defined PET's */
    } pets;
} Gchoice_data3;

```

Gcolr_rep colour representation

```

typedef union {
    Grgb      rgb;        /* Red Green Blue colour specification */
    Gcieluv   cieluv;     /* CIE L*u*v* 1976 colour specification */
    Ghls      hls;        /* Hue Luminance Saturation colour specification */
    Ghsv      hsv;        /* Hue Saturation Value colour specification */
    Gdata      unupp;     /* impl. dep. colour specification */
                /* etc. */
} Gcolr_rep;

```

Gescape_in_data escape input data record

```

typedef union {
    struct Gescape_in_r1 {
        Gdata reg_dep; /* reg. dep. */
    } escape_r1; /* escape 1 data */
    struct Gescape_in_u1 {
        Gdata impl_dep; /* impl. dep. */
    } escape_u1; /* escape -1 data */
    /* etc. */
} Gescape_in_data;

```

Gescape_out_data escape output data record

```

typedef union {
    struct Gescape_out_r1 {
        Gdata reg_dep; /* reg. dep. */
    } escape_r1; /* escape 1 data */
    struct Gescape_out_u1 {
        Gdata impl_dep; /* impl. dep. */
    } escape_u1; /* escape -1 data */
    /* etc. */
} Gescape_out_data;

```

Ggdp_data gdp data record

```

typedef union {
    struct Ggdp_r1 {
        Gdata reg_dep; /* reg. dep. */
    } gdp_r1; /* gdp 1 data */
    struct Ggdp_u1 {
        Gdata impl_dep; /* impl. dep. */
    } gdp_u1; /* gdp -1 data */
    /* etc. */
} Ggdp_data;

```

Ggdp_data3 gdp data record 3

```

typedef union {
    struct Ggdp3_r1 {
        Gdata reg_dep; /* reg. dep. */
    } gdp3_r1; /* gdp 1 data */
    struct Ggdp3_u1 {
        Gdata impl_dep; /* impl. dep. */
    } gdp3_u1; /* gdp -1 data */
    /* etc. */
} Ggdp_data3;

```

Type Definitions

Implementation Dependent Type Definitions

Gitem_data item data record

```

typedef struct {
    Gint          type;          /* item type                */
    Gint          length;       /* item data record length  */
    union {
        Gctrl_flag clear_ws;    /* control flag             */
        Gupd_regen_flag upd_ws; /* regen. flag              */
        struct Gdefer_st {
            Gdefer_mode defer_mode; /* deferral mode          */
            Girg_mode   irg_mode;   /* irg mode                */
        } defer_st;
        /* etc. */
        Gdata gdp_unsupp; /* GDPs not supported by impl. */
        Gdata impl_dep;  /* impl. dependent            */
    } data;
} Gitem_data;

```

Implementation Dependent Type Definitions

Type Definitions

Gloc_data locator data record

```

typedef struct {
    union Gloc_pets {
        struct Gloc_pet_r1 {
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gloc_pet_r2 {
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gloc_pet_r3 {
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gloc_pet_r4 {
            Gattr_ctrl_flag
            /* attribute
            Gline_attrs
            /* polyline
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r4; /* pet 4 data */
        struct Gloc_pet_r5 {
            Gattr_ctrl_flag
            /* attribute
            Gline_fill_ctrl_flag
            /* polyline/fill area
            union Gloc_attrs {
                Gline_attrs    line_attrs;
                /* polyline    attrs. */
                Gfill_attrs    fill_attrs;
                /* fill area   attrs. */
            } attrs;
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r5; /* pet 5 data */
        struct Gloc_pet_r6 {
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_r6; /* pet 6 data */
        struct Gloc_pet_ul {
            Gdata
            impl_dep;      /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* . . . impl. defined PET's */
    } pets;
} Gloc_data;

```

Type Definitions

Implementation Dependent Type Definitions

Gloc_data3 locator data record 3

```

typedef struct {
    union Gloc3_pets {
        struct Gloc3_pet_r1 {
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gloc3_pet_r2 {
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gloc3_pet_r3 {
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gloc3_pet_r4 {
            Gattr_ctrl_flag                      attr_ctrl_flag;
            /* attribute                          control flag */
            Gline_attrs                          line_attrs;
            /* polyline                          attributes */
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r4; /* pet 4 data */
        struct Gloc3_pet_r5 {
            Gattr_ctrl_flag                      attr_ctrl_flag;
            /* attribute                          control flag */
            Gline_fill_ctrl_flag                 line_fill_ctrl_flag;
            /* polyline/fill area                 control flag */
            union Gloc3_attrs {
                Gline_attrs                      line_attrs;
                /* polyline                      attrs. */
                Gfill_attrs                      fill_attrs;
                /* fill area                     attrs. */
            } attrs;
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r5; /* pet 5 data */
        struct Gloc3_pet_r6 {
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_r6; /* pet 6 data */
        struct Gloc3_pet_ul {
            Gdata                                impl_dep;        /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* . . . impl. defined PET's */
    } pets;
} Gloc_data3;

```

Gpick_data pick data record

```

typedef struct {
    union Gpick_pets {
        struct Gpick_pet_r1 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gpick_pet_r2 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gpick_pet_r3 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gpick_pet_ul {
            Gdata impl_dep; /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* etc. */
    } pets;
} Gpick_data;

```

Gpick_data3 pick data record 3

```

typedef struct {
    union Gpick3_pets {
        struct Gpick3_pet_r1 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gpick3_pet_r2 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gpick3_pet_r3 {
            Gdata impl_dep; /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gpick3_pet_ul {
            Gdata impl_dep; /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* etc. */
    } pets;
} Gpick_data3;

```

Type Definitions

Implementation Dependent Type Definitions

Gstring_data string data record

```
typedef struct {
    Gint  in_buf_size; /* input buffer size (nr. of bytes) */
    Gint  init_pos;    /* initial [editing] position */
    union string_pets {
        struct Gstring_pet_r1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gstring_pet_ul {
            Gdata  impl_dep; /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* etc. */
    } pets;
} Gstring_data;
```

Gstring_data3 string data record 3

```
typedef struct {
    Gint  in_buf_size; /* input buffer size (nr. of bytes) */
    Gint  init_pos;    /* initial [editing] position */
    union string3_pets {
        struct Gstring3_pet_r1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gstring3_pet_ul {
            Gdata  impl_dep; /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* etc. */
    } pets;
} Gstring_data3;
```

Implementation Dependent Type Definitions

Type Definitions

Gstroke_data stroke data record

```

typedef struct {
    Gint    in_buf_size;    /* input buffer size (nr. of points) */
    Gint    init_pos;      /* initial [editing] position */
    Gfloat  x_interval;    /* X interval */
    Gfloat  y_interval;    /* Y interval */
    Gfloat  time_interval; /* time interval */
    union Gstroke_pets {
        struct Gstroke_pet_r1 {
            Gdata    impl_dep; /* impl. dep. */
        } pet_r1;          /* pet 1 data */
        struct Gstroke_pet_r2 {
            Gdata    impl_dep; /* impl. dep. */
        } pet_r2;          /* pet 2 data */
        struct Gstroke_pet_r3 {
            Gattr_ctrl_flag attr_ctrl_flag;
            /* attribute control flag */
            Gmarker_attrs  marker_attrs;
            /* polymarker attrs. */
            Gdata    impl_dep; /* impl. dep. */
        } pet_r3;          /* pet 3 data */
        struct Gstroke_pet_r4 {
            Gattr_ctrl_flag attr_ctrl_flag;
            /* attribute control flag */
            Gline_attrs  line_attrs;
            /* polyline attrs. */
            Gdata    impl_dep; /* impl. dep. */
        } pet_r4;          /* pet 4 data */
        struct Gstroke_pet_u1 {
            Gdata    impl_dep; /* impl. dep. */
        } pet_u1;          /* pet -1 data */
        /* etc. */
    } pets;
} Gstroke_data;

```

Type Definitions

Implementation Dependent Type Definitions

Gstroke_data3 stroke data record 3

```

typedef struct {
    Gint    in_buf_size;    /* input buffer size (nr. of points) */
    Gint    init_pos;      /* initial [editing] position */
    Gfloat  x_interval;    /* X interval */
    Gfloat  y_interval;    /* Y interval */
    Gfloat  z_interval;    /* Z interval */
    Gfloat  time_interval; /* time interval */
    union Gstroke3_pets {
        struct Gstroke3_pet_r1 {
            Gdata    impl_dep; /* impl. dep. */
        } pet_r1; /* pet 1 data */
        struct Gstroke3_pet_r2 {
            Gdata    impl_dep; /* impl. dep. */
        } pet_r2; /* pet 2 data */
        struct Gstroke3_pet_r3 {
            Gattr_ctrl_flag attr_ctrl_flag;
            /* attribute control flag */
            Gmarker_attrs marker_attrs;
            /* polymarker attrs. */
            Gdata    impl_dep; /* impl. dep. */
        } pet_r3; /* pet 3 data */
        struct Gstroke3_pet_r4 {
            Gattr_ctrl_flag attr_ctrl_flag;
            /* attribute control flag */
            Gline_attrs line_attrs;
            /* polyline attrs. */
            Gdata    impl_dep;
        } pet_r4; /* pet 4 data */
        struct Gstroke3_pet_ul {
            Gdata    impl_dep; /* impl. dep. */
        } pet_ul; /* pet -1 data */
        /* etc. */
    } pets;
} Gstroke_data3;

```

Implementation Dependent Type Definitions

Type Definitions

Gval_data valuator data record

```

typedef struct {
    Gfloat  low_value;    /* low value */
    Gfloat  high_value;   /* high value */
    union Gval_pets {
        struct Gval_pet_r1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r1;    /* pet 1 data */
        struct Gval_pet_r2 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r2;    /* pet 2 data */
        struct Gval_pet_r3 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r3;    /* pet 3 data */
        struct Gval_pet_u1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_u1;    /* pet -1 data */
        /* etc. */
    } pets;
} Gval_data;

```

Gval_data3 valuator data record 3

```

typedef struct {
    Gfloat  low_value;    /* low value */
    Gfloat  high_value;   /* high value */
    union Gval3_pets {
        struct Gval3_pet_r1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r1;    /* pet 1 data */
        struct Gval3_pet_r2 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r2;    /* pet 2 data */
        struct Gval3_pet_r3 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_r3;    /* pet 3 data */
        struct Gval3_pet_u1 {
            Gdata  impl_dep; /* impl. dep. */
        } pet_u1;    /* pet -1 data */
        /* etc. */
    } pets;
} Gval_data3;

```

5.4 Implementation Independent Type Definitions

REMARKS:

- 1) For some two-field enumeration types, the order of the fields has been reversed with respect to ISO 8805; this reversal has been done in order to keep numerical compatibility with ISO/IEC 8806-1 (GKS-3D/ FORTRAN); the enumeration types are: