

# Information technology — Computer graphics — Graphical Kernel Systems (GKS) language bindings —

## Part 4: C

The European Standard EN ISO/IEC 8651-4:1996 has the status of a British Standard

ICS 35.060; 35.140

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW



Document Sharing Hub

# National foreword

This British Standard is the English language version of EN ISO/IEC 8651-4:1996. It is identical with ISO/IEC 8651-4:1995. It supersedes BS ISO/IEC 8651-4:1991 which is withdrawn.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.

### 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 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.**

### Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the EN ISO/IEC title page, page 2, the ISO/IEC title page, pages ii to vi, pages 1 to 295 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

### Amendments issued since publication

| Amd. No. | Date | Comments |
|----------|------|----------|
|          |      |          |
|          |      |          |
|          |      |          |
|          |      |          |
|          |      |          |

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Board and comes into effect on 15 April 1997

© BSI 05-2000

ISBN 0 580 27237 0

# Contents

|                        | Page               |
|------------------------|--------------------|
| National foreword      | Inside front cover |
| Foreword               | 2                  |
| Foreword               | v                  |
| Text of ISO/IEC 8651-4 | 1                  |



English version

# Information technology — Computer graphics — Graphical Kernel System (GKS) language bindings — Part 4: C

(ISO/IEC 8651-4:1995)

Technologies de l'information — Infographie —  
Interfaces langage avec GKS — Partie 4: C  
(ISO/IEC 8651-4:1995)

Informationstechnik — Verarbeitung  
graphischer Daten — Sprachbindungen für das  
graphische Kernsystem (GKS) — Teil 4: C  
(ISO/IEC 8651-4:1995)

This European Standard was approved by CEN on 1996-04-18. CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

The European Standards exist in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

## CEN

European Committee for Standardization  
Comité Européen de Normalisation  
Europäisches Komitee für Normung

**Central Secretariat: rue de Stassart 36, B-1050 Brussels**

## Foreword

The text of the International Standard from ISO/IEC/JTC 1 “Information Technology” of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) has been taken over as a European Standard by the Technical Board of CEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by November 1996, and conflicting national standards shall be withdrawn at the latest by November 1996.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this European Standard: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

## Endorsement notice

The text of the International Standard ISO/IEC 8651-4:1995 has been approved by CEN as a European Standard without any modification.

INTERNATIONAL  
STANDARD

**ISO/IEC**  
**8651-4**

Second edition  
1995-06-01

---

---

**Information technology — Computer  
graphics — Graphical Kernel System (GKS)  
language bindings —**

**Part 4:**  
**C**

*Technologies de l'information — Infographie — Interfaces langage avec  
GKS —*

*Partie 4: C*



Reference number  
ISO/IEC 8651-4:1995/EN

**AVA** Documents  
Document Sharing Hub

## Contents

|                                                        | Page |
|--------------------------------------------------------|------|
| Foreword                                               | v    |
| Introduction                                           | 1    |
| 1 Scope                                                | 1    |
| 2 Normative references                                 | 1    |
| 3 The C language binding                               | 1    |
| 3.1 Classification and designation                     | 1    |
| 3.2 Functions versus macros                            | 1    |
| 3.3 Character strings                                  | 1    |
| 3.4 Function identifiers                               | 2    |
| 3.5 Registration                                       | 2    |
| 3.6 Identifiers for graphical items                    | 2    |
| 3.7 Return values                                      | 2    |
| 3.8 Headers                                            | 2    |
| 3.8.1 <b>gks.h</b>                                     | 2    |
| 3.8.2 <b>gks_compat.h</b>                              | 3    |
| 3.9 Memory management                                  | 3    |
| 3.9.1 Functions which return simple lists              | 3    |
| 3.9.2 Functions which return complex data structures   | 3    |
| 3.10 Error handling                                    | 5    |
| 3.10.1 Application supplied error handlers             | 5    |
| 3.10.2 Error codes                                     | 5    |
| 3.10.3 C-specific GKS errors                           | 5    |
| 3.11 Colour representations and specifications         | 5    |
| 3.12 Colour characteristics                            | 5    |
| 3.13 Storage of multi-dimensional arrays               | 5    |
| 3.13.1 Storage of 2*3 matrices                         | 5    |
| 3.13.2 Storage of conics in 3*3 matrices               | 5    |
| 3.13.3 Storage of colour arrays                        | 6    |
| 3.14 Compatibility with the 1991 edition               | 6    |
| 4 Tables                                               | 6    |
| 4.1 Abbreviation policy in construction of identifiers | 6    |
| 4.2 Table of abbreviations used                        | 7    |
| 4.3 Function names                                     | 10   |
| 4.3.1 List ordered alphabetically by bound name        | 10   |
| 4.3.2 List ordered alphabetically by GKS name          | 16   |
| 5 Type definitions                                     | 24   |
| 5.1 Mapping of GKS data types                          | 24   |
| 5.2 Environment-defined type definitions               | 24   |
| 5.3 Implementation dependent type definitions          | 25   |
| 5.4 Implementation independent type definitions        | 31   |
| 6 Macro definitions                                    | 77   |
| 6.1 Function identifiers                               | 77   |
| 6.1.1 In order of appearance                           | 77   |
| 6.1.2 In alphabetical order                            | 81   |
| 6.2 Error codes                                        | 85   |
| 6.3 Miscellaneous                                      | 88   |
| 6.3.1 Linetypes                                        | 88   |
| 6.3.2 Marker types                                     | 88   |
| 6.3.3 Hatch styles                                     | 89   |
| 6.3.4 Colour models                                    | 89   |

|                       | Page                                                           |
|-----------------------|----------------------------------------------------------------|
| 6.3.5                 | Prompt and echo types 89                                       |
| 6.3.6                 | Default parameter of <b>gopen_gks</b> 89                       |
| 7                     | C GKS function interface 89                                    |
| 7.1                   | Notational conventions 89                                      |
| 7.2                   | Workstation independent functions 90                           |
| 7.2.1                 | Control functions 90                                           |
| 7.2.2                 | Output functions 91                                            |
| 7.2.3                 | Design output functions 93                                     |
| 7.2.4                 | Primitive attribute functions 95                               |
| 7.2.5                 | Normalization transformation functions 99                      |
| 7.2.6                 | NDC picture functions 100                                      |
| 7.2.7                 | Metafile functions 100                                         |
| 7.2.8                 | Picture part store functions 101                               |
| 7.2.9                 | Input functions 102                                            |
| 7.2.10                | Font and glyph functions 107                                   |
| 7.2.11                | Audit and playback functions 107                               |
| 7.2.12                | Inquiry functions 108                                          |
| 7.2.13                | Utility functions 117                                          |
| 7.3                   | Workstation functions 119                                      |
| 7.3.1                 | Control functions 119                                          |
| 7.3.2                 | Inquiry functions 123                                          |
| 7.3.3                 | Retrieval functions 135                                        |
| 7.3.4                 | Viewing utility functions 135                                  |
| 7.3.5                 | Colour utility functions 135                                   |
| 7.4                   | Segment functions and workstation activation functions 135     |
| 7.4.1                 | Segment functions 136                                          |
| 7.4.2                 | Workstation activation functions 137                           |
| 7.4.3                 | Utility functions 137                                          |
| Annex A (informative) | Compiled GKS/C specification 139                               |
| A.1                   | Data types in compilation order 139                            |
| A.2                   | Macros 185                                                     |
| A.3                   | Function calls 193                                             |
| A.4                   | Compatibility layer 222                                        |
| Annex B (informative) | Sample programs 233                                            |
| B.1                   | STAR 233                                                       |
| B.2                   | IRON 235                                                       |
| Annex C (informative) | Short function identifiers 241                                 |
| C.1                   | In order of appearance 241                                     |
| C.2                   | In alphabetical order 247                                      |
| Annex D (informative) | Memory management 254                                          |
| D.1                   | Introduction 254                                               |
| D.2                   | Functions that return simple lists 254                         |
| D.2.1                 | Operation of <b>ginq_list_line_inds</b> 255                    |
| D.3                   | Functions that return structured data 257                      |
| D.3.1                 | Operation of <b>gcreate_store</b> 258                          |
| D.3.2                 | Operation of <b>ginq_stroke_st</b> and <b>ginq_pat_rep</b> 260 |
| D.3.3                 | Operation of <b>gdel_store</b> 263                             |

|                                                                                                        | Page |
|--------------------------------------------------------------------------------------------------------|------|
| Annex E (informative) Compatibility with the 1991 edition of ISO/IEC 86514                             | 265  |
| E.1 Comparison of this edition of ISO/IEC 86514 with the 1991 edition                                  | 265  |
| E.1.1 Changes in ISO/IEC 86514 data types                                                              | 265  |
| E.1.2 Changes in ISO/IEC 86514 functions                                                               | 266  |
| E.2 The compatibility layer                                                                            | 267  |
| E.3 The header <b>gks_compat.h</b>                                                                     | 267  |
| E.4 Data types in <b>gks_compat.h</b>                                                                  | 267  |
| E.4.1 Renamed data types                                                                               | 267  |
| E.4.2 Renamed fields of data types                                                                     | 268  |
| E.4.3 Obsolete data types                                                                              | 268  |
| E.5 Macros                                                                                             | 271  |
| E.6 Functions in the compatibility layer                                                               | 272  |
| E.6.1 Replaced functions                                                                               | 272  |
| E.6.2 Obsolete functions                                                                               | 274  |
| Annex F (informative) Function lists                                                                   | 280  |
| F.1 Alphabetic by GKS name                                                                             | 280  |
| F.2 Alphabetic by binding name                                                                         | 287  |
| Figure D.1 — Sample program for the demonstration of the memory management for simple lists            | 254  |
| Figure D.2 — Values of parameters of <code>ginq_list_line_inds</code> at start of call                 | 255  |
| Figure D.3 — Values of parameters at end of call in case 1                                             | 256  |
| Figure D.4 — Values of parameters at end of call in case 2                                             | 258  |
| Figure D.5 — Sample program for the demonstration of the memory management for complex structured data | 258  |
| Figure D.6 — Values of parameters of <code>gcreate_store</code> at start of call                       | 259  |
| Figure D.7 — Values of parameters of <code>gcreate_store</code> at end of successful call              | 259  |
| Figure D.8 — Values of parameters of <code>gind_stroke_st</code> at start of call                      | 260  |
| Figure D.9 — Values of parameters of <code>gind_stroke_st</code> at end of call                        | 261  |
| Figure D.10 — Values of parameters of <code>gind_pat_rep</code> at start of call                       | 262  |
| Figure D.11 — Values of parameters of <code>gind_pat_rep</code> at end of call                         | 263  |
| Figure D.12 — Values of parameters of <code>gdel_store</code> at start of call                         | 264  |
| Figure D.13 — Values of parameters of <code>pf gdel_store</code> at end of successful call             | 264  |
| Figure D.14 — Values of parameters of <code>pf gdel_store</code> at end of unsuccessful call           | 265  |
| Table E.1 — Position of compatibility layer                                                            | 267  |

**Descriptors:** Data processing, graphic data processing, information interchange, computer software, computer interfaces, programming languages, C (programming language).

## Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8651-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics and image processing*.

This second edition cancels and replaces the first edition (ISO/IEC 8651-4:1991), which has been technically revised.

ISO/IEC 8651-4 consists of the following parts, under the general title *Information technology — Computer graphics — Graphical Kernel System (GKS) language bindings*:

- *Part 1: FORTRAN*;
- *Part 2: Pascal*;
- *Part 3: Ada*;
- *Part 4: C*.

Annex A to Annex F of this part of ISO/IEC 8651 are for information only.



## Introduction

The Graphical Kernel System (GKS) functional description is registered as ISO/IEC 7942-1:1994. As explained in the Scope and Field of Application of ISO/IEC 7942-1, 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 8651 is to define a standard binding for the C computer programming language.

## 1 Scope

The Graphical Kernel System (GKS), ISO/IEC 7942-1:1994, specifies a language independent nucleus of a graphics system. For integration into a programming language, GKS is embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 8651 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 provisions of this part of ISO/IEC 8651. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 8651 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/IEC 7942-1:1994, *Information technology — Computer graphics and image processing — Graphical Kernel System (GKS) — Part 1: Functional description*.

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

ISO/IEC TR 9973:1994, *Information technology — Computer graphics and image processing — Procedures for registration of graphical Items*.

## 3 The C language binding

The C language binding of GKS shall be as described in clauses 3 to 7.

### 3.1 Classification and designation

This part of ISO/IEC 8651 incorporates the rules of conformance defined in the GKS Standard (ISO/IEC 7942-1) for GKS implementations, with those additional requirements specifically defined for C bindings in GKS.

The following criteria shall determine conformance of an implementation to this part of ISO/IEC 8651:

In order to conform, an implementation of the C binding of GKS shall implement GKS as specified in ISO/IEC 7942-1. It shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 8651 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 are all mapped to C functions which begin with the letter **g**. Words and phrases used in the GKS 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 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 language (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 object-like macros in the header which equate the long names to a set of short names. A possible set of short names for a compiler that accepts only 8 characters for external definitions may be found in Annex C.

### 3.5 Registration

ISO/IEC 7942 reserves certain value ranges for registration<sup>1)</sup> as graphical items. The registered graphical items will be bound to the C programming language (and other programming languages). The registered item binding will be consistent with the binding presented in this part of ISO/IEC 8651.

### 3.6 Identifiers for graphical items

Generalized Drawing Primitives and Escape functions are referenced via identifiers. This part of ISO/IEC 8651 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** 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)      /* where 'n' is the registered GDP identifier */
```

The format for unregistered GDP identifiers is:

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

The format for registered Escape function identifiers is:

```
#define      GESCAPE_Rn   (n)      /* where 'n' is the registered Escape identifier */
```

The format for unregistered Escape function identifiers is:

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

### 3.7 Return values

All GKS/C functions have return value **void**.

### 3.8 Headers

#### 3.8.1 *gks.h*

C provides a mechanism to access information stored in a header via the **#include** preprocessing directive. Clause 5 of this part of ISO/IEC 8651 describes the data types that shall be defined in the header **gks.h** which shall be included in any application program that intends to use GKS via the C binding.

This part of ISO/IEC 8651 uses the data type **size\_t** (inter alia as a field in the data type **Gdata**). The type **size\_t** is environment-defined (i.e. **unsigned long** or **unsigned int**) and is defined in the headers **<stdio.h>**, **<stddef.h>**, **<stdlib.h>**, **<string.h>**, **<time.h>**.

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

---

<sup>1)</sup> For the purpose of this part of ISO/IEC 8651 and according to the rules for the designation and operation of registration authorities in the ISO/IEC Directives, the ISO/IEC council has designated the National Institute of Standards and Technology (Institute of Computer Sciences and Technology), A-266 Technology Building, Gaithersburg, MD 20899, USA to act as registration authority.

The header **gks.h** shall also contain external declarations for all GKS/C functions because they have a **void** return type. For example, the declaration for the function **gopen\_gks** would look like this:

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

### 3.8.2 *gks\_compat.h*

For application programs which used to run on top of the 1991 edition of this part of ISO/IEC, the header **gks\_compat.h** is provided. **gks\_compat.h** includes GKS/C data types that are no longer supported, as well as external declarations for all GKS/C functions that are no longer supported. Implementations of this part of ISO/IEC 8651 shall support these functions in a compatibility layer, according to the guidelines in Annex G of ISO/IEC 7942-1:1994.

## 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 LIST OF MARKER INDICES returns a list of the available marker indices. The other class returns complex, heterogeneous data structures. For example, the function INQUIRE GKS STATE LIST ENTRY returns a piece of the GKS state which may include several data structures of different length. 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.

### 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 D, a possible underlying mechanism is described.

### 3.9.2 *Functions which return complex data structures*

The data returned by inter alia the ESCAPE function, the AWAIT INPUT function and the functions which return state lists or description 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 8651 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 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 8651 defines two errors that can occur when using or creating a *Store*; these errors are described in 6.2. 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

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 error indicator is set to one of the following error values:

|                    |                                                                 |
|--------------------|-----------------------------------------------------------------|
| GE_GKS_NOT_OPEN    | GKS not in proper state;<br>GKS shall be in the state (GKOP, *) |
| GE_ERR_ALLOC_STORE | Error while allocating Store                                    |

**Errors:** None.

DELETE STORE

Parameters:

|     |                 |       |
|-----|-----------------|-------|
| Out | error indicator | I     |
| 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:

|                 |                                                                 |
|-----------------|-----------------------------------------------------------------|
| GE_GKS_NOT_OPEN | GKS not in proper state;<br>GKS shall be in the state (GKOP, *) |
|-----------------|-----------------------------------------------------------------|

**Errors:** None.

In 7.2.13.2, the C specification of these functions is given. In Annex D, 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 7.2.13.2)

#### SET ERROR HANDLER

Parameters:

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

**Effect:** Sets the GKS 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/IEC 7942 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 8651 defines a set of constants for the GKS error numbers. Each error constant begins with the characters **GE\_**. See also 6.2 for the error macros.

#### 3.10.3 C-specific GKS errors

This part of ISO/IEC 8651 defines some additional error messages. In 6.2 the numbers and their macros are given.

### 3.11 Colour representations and specifications

GKS 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**, **Gcielv**, **Ghsv**, **Ghls**). The colour representation and specification are defined in 5.4 by the types **Gcolr\_rep** and **Gcolr\_specif**.

#### 3.12 Colour characteristics

GKS defines the colour characteristics as a basic type. The colour characteristics is defined in this part of ISO/IEC 8651 by the data type **Gcolr\_chars**, which is documented in 5.3.

### 3.13 Storage of multi-dimensional arrays

#### 3.13.1 Storage of 2\*3 matrices

The entries of **Gtran\_matrix** data types shall be stored such that the segment transformation is defined by

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

where **p** is a 2D point, **Tp** its transformation and **mat** is of type **Gtran\_matrix**.

#### 3.13.2 Storage of conics in 3\*3 matrices

The entries of **Gconic\_matrix** data types shall be stored such that the conic is defined by

$$\begin{aligned} &a[0,0]x^2 + (a[0,1] + a[1,0])xy + a[1,1]y^2 \\ &+ (a[0,2] + a[2,0])x + (a[1,2] + a[2,1])y + a[2,2] = 0 \end{aligned}$$

where **a** is of type **Gconic\_matrix**.

### 3.13.3 Storage of colour arrays

The entries of **Gpat\_rep** data types shall be stored such that the colour specifier at the (i,j)-th entry is given by

```

xxxi,j   =   colr_rect.dir_colr.xxx [i + DX*j]; i = 0,..., DX - 1; j
colr_indi,j =   colr_rect.colr_array [i + DX*j];
DX       =   colr_rect.dims.size_x;
DY       =   colr_rect.dims.size_y;

```

where xxx = (rgb | cielv | hls | hsv) is of type **Gxxx** and **colr\_rect** is of type **Gpat\_rep**.

### 3.14 Compatibility with the 1991 edition

In the second (1994) edition of ISO/IEC 7942 some functions and data types of the 1985 edition have been overtaken or deprecated. They are documented in the informative Annex G. Examples are the GKSM functions, which have been overtaken by AUDIT functions.

In this edition of ISO/IEC 8651-4 *all* functions and *all* data types of the 1991 edition (the C binding of the 1985 edition of ISO/IEC 7942) have been preserved. The support of overtaken or deprecated functions or data types falls into two categories:

- 1) *Functions which have been renamed (e.g. INQUIRE TEXT EXTENT → GET TEXT EXTENT) or which have been incorporated into more compact functions (e.g. the OUTPUT functions).*

These functions and their related data types are documented in clauses 5 and 7. For example, the function CREATE OUTPUT PRIMITIVE has been bound to the C function **gcreate\_out\_prim**.

Besides that, however, the output functions **gpolyline**, **gpolyline\_set**, **gnurb\_set**, **gconic\_sec\_set**, **gpolymarker**, **gfill\_area**, **gfill\_area\_set**, **gell\_sec\_set**, **gell\_seg\_set**, **gell\_disc\_set**, **gclosed\_nurb\_set**, **gtext**, **gcell\_array**, **gdesign**, **ggdp** have been defined.

- 2) *Functions which have been deprecated (INQUIRE GKS LEVEL) or which have been overtaken by other functions (GKSM functions, for example).*

These functions are documented in Annex E. They will be deleted at the next edition of this part of ISO/IEC 8651.

## 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.
- 4) The word REALIZED is not abbreviated in the second field of the enumeration data type **Ginq\_type**; in all other cases it is abbreviated using the list in 4.2.
- 5) Construction of GKS/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 linewidth in the field of **Gline\_bundle** is abbreviated to **width**, because the context makes clear which width is used;
  - d) Function macros:
 

“Gfn\_” followed by abbreviation of function name;

e) Error macros:

“GE\_” followed by some abbreviated expression;

f) 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.

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

| Word or Phrase            | Abbreviation |
|---------------------------|--------------|
| CIE L*u*v* (colour model) | cieluv       |
| accumulate                | accum        |
| actual                    | act          |
| addition                  | add          |
| alignment                 | align        |
| allocate                  | alloc        |
| and                       | NULL         |
| application               | appl         |
| arithmetic                | arith        |
| associate(d)              | assoc        |
| at                        | NULL         |
| attribute                 | attr         |
| attribute source flag     | asf          |
| availability              | avail        |
| available                 | avail        |
| begin                     | beg          |
| between                   | NULL         |
| boundaries                | bndries      |
| boundary                  | bndry        |
| buffer                    | buf          |
| cannot                    | cant         |
| centre                    | ctr          |
| character                 | char         |
| characteristics           | chars        |
| circular                  | circ         |
| classification            | class        |
| clipping                  | clip         |
| colour                    | colr         |
| component                 | comp         |
| composite                 | comp         |
| composition               | comp         |
| concatenation             | concat       |
| connection                | conn         |
| contour                   | cont         |
| control                   | ctrl         |
| conversion                | conv         |
| coordinate                | coord        |
| correspondence            | corr         |
| criterion                 | crit         |
| current                   | cur          |
| dashed                    | dash         |
| default                   | def          |
| defined                   | def          |
| delete                    | del          |
| deletion                  | del          |
| dependent                 | dep          |
| description table         | dt           |

## Word or Phrase

designation  
detectability  
detectable  
device  
device coordinate(s)  
digital  
dimension  
direction  
display  
dotted  
duplicate  
elliptic  
equal  
error  
evaluate  
expansion  
facility  
factor  
fill area  
from  
function  
generalized drawing primitive  
generic  
gks closed  
gks open  
graphical  
handling  
height  
highlighted  
highlighting  
homogeneous  
horizontal  
hue lightness saturation (colour model)  
hue saturation value (colour model)  
hyperbola  
identifier  
implementation  
implicit  
in  
in use  
index  
indicator  
individual  
initial  
initialize  
input  
input/output  
inquire  
instance  
integer  
interior  
invalid  
invisible  
length  
less than  
less than or equal  
library  
locator  
logical  
luminance  
mapped  
mapping  
maximum  
measure  
memory

## Abbreviation

design  
det  
det  
dev  
dc  
digit  
dim  
dir  
disp  
dot  
dup  
ell  
eq  
err  
eval  
expan  
fac  
NULL  
fill  
NULL  
func  
gdp  
gen  
gkcl  
gkop  
graph  
hand  
ht  
highl  
highl  
homo  
hor  
hls  
hsv  
hyp  
id  
impl  
impl  
NULL  
NULL  
ind  
ind  
indiv  
init  
init  
in  
io  
inq  
inst  
int  
int  
inval  
invis  
NULL  
lt  
le  
lib  
loc  
NULL  
lum  
map  
map  
max  
meas  
mem

| Word or Phrase                  | Abbreviation |
|---------------------------------|--------------|
| metafile                        | mf           |
| minimum                         | min          |
| monochrome                      | monochr      |
| nominal                         | nom          |
| normal                          | norm         |
| normalization                   | norm         |
| normalized device coordinate(s) | ndc          |
| number                          | num          |
| of                              | NULL         |
| on                              | NULL         |
| operating                       | op           |
| operation                       | op           |
| orientation                     | ori          |
| output                          | out          |
| overflow                        | overf        |
| parallelogram                   | paral        |
| parameter                       | param        |
| pattern                         | pat          |
| picture                         | pic          |
| picture part closed             | ppcl         |
| picture part open               | ppop         |
| pointer                         | ptr          |
| polynomial                      | pol          |
| position                        | pos          |
| precision                       | prec         |
| predefined                      | pred         |
| primary                         | prim         |
| primitive                       | prim         |
| priority                        | pri          |
| process                         | proc         |
| prompt                          | pr           |
| prompt and echo type            | pet          |
| queue                           | NULL         |
| rational                        | rat          |
| realized                        | real         |
| record                          | rec          |
| rectangle                       | rect         |
| red green blue (colour model)   | rgb          |
| reference                       | ref          |
| registered                      | reg          |
| reject                          | rej          |
| relative                        | rel          |
| replication                     | repl         |
| representation                  | rep          |
| request(ed)                     | req          |
| retrieve                        | ret          |
| rubber                          | rub          |
| saturation                      | satur        |
| scale factor                    | NULL         |
| section                         | sec          |
| sector                          | sec          |
| segment                         | seg          |
| segment closed                  | sgcl         |
| segment open                    | sgop         |
| select                          | sel          |
| selection                       | sel          |
| selector                        | sel          |
| sequence                        | seq          |
| source                          | src          |
| spacing                         | space        |
| specification                   | specif       |
| specified                       | specif       |
| specifier                       | specif       |
| state                           | st           |
| state list                      | sl           |

## Word or Phrase

stencil closed  
stencil open  
storage  
supported  
suppress  
tiling closed  
tiling open  
to  
tracking  
transform(ation)  
unavailable  
undefined  
undetectable  
unregistered  
used  
valuator  
vector  
vertical  
viewport  
visibility  
visible  
visual  
which  
window  
with  
workstation  
world coordinate

## Abbreviation

stcl  
stop  
store  
NULL  
suppr  
tlcl  
tlop  
NULL  
track  
tran  
unavail  
undef  
undet  
u  
NULL  
val  
vec  
vert  
vp  
vis  
vis  
vis  
NULL  
win  
NULL  
ws  
wc

## 4.3 Function names

### 4.3.1 List ordered alphabetically by bound name

|                                      |                                                  |
|--------------------------------------|--------------------------------------------------|
| <b>gaccum_tran_matrix</b>            | ACCUMULATE TRANSFORMATION MATRIX                 |
| <b>gactivate_ws</b>                  | ACTIVATE WORKSTATION                             |
| <b>gadd_set_names_nameset</b>        | ADD SET OF NAMES TO NAMESET                      |
| <b>gadd_set_names_ndc_pic</b>        | ADD SET OF NAMES TO NDC PICTURE                  |
| <b>gadd_set_scissors_ndc_pic</b>     | ADD SET OF SCISSORS TO NDC PICTURE               |
| <b>gadd_set_scissors_scissor_set</b> | ADD SET OF SCISSORS TO SCISSOR SET               |
| <b>gappend_pic_part</b>              | APPEND PICTURE PART                              |
| <b>gar_pic_part</b>                  | ARCHIVE PICTURE PART                             |
| <b>gassoc_seg_ws</b>                 | ASSOCIATE SEGMENT WITH WORKSTATION               |
| <b>gaudit</b>                        | AUDIT                                            |
| <b>gawait_event</b>                  | AWAIT EVENT                                      |
| <b>gawait_in</b>                     | AWAIT INPUT                                      |
| <b>gcell_array</b>                   | CELL ARRAY                                       |
| <b>gclear_ws</b>                     | CLEAR WORKSTATION                                |
| <b>gclose_gks</b>                    | CLOSE GKS                                        |
| <b>gclose_pic_part</b>               | CLOSE PICTURE PART                               |
| <b>gclose_seg</b>                    | CLOSE SEGMENT                                    |
| <b>gclose_stencil</b>                | CLOSE STENCIL                                    |
| <b>gclose_tiling</b>                 | CLOSE TILING                                     |
| <b>gclose_ws</b>                     | CLOSE WORKSTATION                                |
| <b>gclosed_nurb_set</b>              | CLOSED NURB SET                                  |
| <b>gconic_sec_set</b>                | CONIC SECTION SET                                |
| <b>gconv_colr</b>                    | CONVERT COLOUR                                   |
| <b>gcopy_blank_real_pic_real_mf</b>  | COPY BLANK REALIZED PICTURE TO REALIZED METAFILE |
| <b>gcopy_ndc_mf_pic_ndc_pic</b>      | COPY NDC METAFILE PICTURE TO NDC PICTURE         |
| <b>gcopy_ndc_pic_ndc_mf</b>          | COPY NDC PICTURE TO NDC METAFILE                 |
| <b>gcopy_ndc_pic_pic_part_store</b>  | COPY NDC PICTURE TO PICTURE PART STORE           |
| <b>gcopy_pic_part_pic_part_store</b> | COPY PICTURE PART FROM PICTURE PART STORE        |

|                                   |                                            |
|-----------------------------------|--------------------------------------------|
| <b>gcopy_real_mf_pic_backdrop</b> | COPY REALIZED METAFILE PICTURE TO BACKDROP |
| <b>gcopy_real_pic_real_mf</b>     | COPY REALIZED PICTURE TO REALIZED METAFILE |
| <b>gcopy_seg_ws</b>               | COPY SEGMENT FROM WORKSTATION              |
| <b>gcreate_out_prim</b>           | CREATE OUTPUT PRIMITIVE                    |
| <b>gcreate_seg</b>                | CREATE SEGMENT                             |
| <b>gcreate_stencil_bndry</b>      | CREATE STENCIL FROM BOUNDARY               |
| <b>gcreate_stencil_conts</b>      | CREATE STENCIL FROM CONTOURS               |
| <b>gcreate_store</b>              | CREATE STORAGE                             |
| <b>gcreate_tiling_comp</b>        | CREATE TILING COMPONENT                    |
| <b>gdeactivate_ws</b>             | DEACTIVATE WORKSTATION                     |
| <b>gdef_in_dev</b>                | DEFINE LOGICAL INPUT DEVICE                |
| <b>gdel_pic_part</b>              | DELETE PICTURE PART                        |
| <b>gdel_prims_ndc_pic</b>         | DELETE PRIMITIVES FROM NDC PICTURE         |
| <b>gdel_seg</b>                   | DELETE SEGMENT                             |
| <b>gdel_seg_ws</b>                | DELETE SEGMENT FROM WORKSTATION            |
| <b>gdel_stencil</b>               | DELETE STENCIL                             |
| <b>gdel_store</b>                 | DELETE STORAGE                             |
| <b>gdel_tiling</b>                | DELETE TILING                              |
| <b>gdesign</b>                    | DESIGN                                     |
| <b>gell_disc_set</b>              | ELLIPTIC DISC SET                          |
| <b>gell_sec_set</b>               | ELLIPTIC SECTOR SET                        |
| <b>gell_seg_set</b>               | ELLIPTIC SEGMENT SET                       |
| <b>gemergency_close_gks</b>       | EMERGENCY CLOSE GKS                        |
| <b>gerr_hand</b>                  | ERROR HANDLING                             |
| <b>gerr_log</b>                   | ERROR LOGGING                              |
| <b>gescape</b>                    | ESCAPE                                     |
| <b>geval_circ_arc_3_point</b>     | EVALUATE CIRCULAR ARC 3 POINT              |
| <b>geval_circ_arc_ctr</b>         | EVALUATE CIRCULAR ARC CENTRE               |
| <b>geval_circle</b>               | EVALUATE CIRCLE                            |
| <b>geval_ell</b>                  | EVALUATE ELLIPSE                           |
| <b>geval_ell_arc</b>              | EVALUATE ELLIPTIC ARC                      |
| <b>geval_hyp_arc</b>              | EVALUATE HYPERBOLIC ARC                    |
| <b>geval_par_arc</b>              | EVALUATE PARABOLIC ARC                     |
| <b>geval_tran_matrix</b>          | EVALUATE TRANSFORMATION MATRIX             |
| <b>geval_view_map_matrix</b>      | EVALUATE VIEW MAPPING MATRIX               |
| <b>geval_view_ori_matrix</b>      | EVALUATE VIEW ORIENTATION MATRIX           |
| <b>geval_wc_ord</b>               | EVALUATE WC ORDINATE                       |
| <b>gfill_area</b>                 | FILL AREA                                  |
| <b>gfill_area_set</b>             | FILL AREA SET                              |
| <b>gflush_dev_events</b>          | FLUSH DEVICE EVENTS                        |
| <b>ggdp</b>                       | GENERALIZED DRAWING PRIMITIVE              |
| <b>gget_choice</b>                | GET CHOICE                                 |
| <b>gget_glyph_name</b>            | GET GLYPH NAME                             |
| <b>gget_loc</b>                   | GET LOCATOR                                |
| <b>gget_map_seg_name</b>          | GET MAPPED SEGMENT NAME                    |
| <b>gget_map_ws_id</b>             | GET MAPPED WORKSTATION IDENTIFIER          |
| <b>gget_pick</b>                  | GET PICK                                   |
| <b>gget_stencil_attr</b>          | GET STENCIL ATTRIBUTE                      |
| <b>gget_string</b>                | GET STRING                                 |
| <b>gget_stroke</b>                | GET STROKE                                 |
| <b>gget_text_extent</b>           | GET TEXT EXTENT                            |
| <b>gget_val</b>                   | GET VALUATOR                               |
| <b>gget_ws_status</b>             | GET WORKSTATION STATUS                     |
| <b>ginit_choice</b>               | INITIALIZE CHOICE                          |
| <b>ginit_in_dev</b>               | INITIALIZE LOGICAL INPUT DEVICE            |

|                                      |                                                         |
|--------------------------------------|---------------------------------------------------------|
| <code>ginit_loc</code>               | INITIALIZE LOCATOR                                      |
| <code>ginit_pick</code>              | INITIALIZE PICK                                         |
| <code>ginit_string</code>            | INITIALIZE STRING                                       |
| <code>ginit_stroke</code>            | INITIALIZE STROKE                                       |
| <code>ginit_val</code>               | INITIALIZE VALUATOR                                     |
| <code>ginq_area_facs</code>          | INQUIRE AREA FACILITIES                                 |
| <code>ginq_area_ind</code>           | INQUIRE AREA INDEX                                      |
| <code>ginq_area_rep</code>           | INQUIRE AREA REPRESENTATION                             |
| <code>ginq_asfs</code>               | INQUIRE ATTRIBUTE SOURCE FLAGS                          |
| <code>ginq_char_expan</code>         | INQUIRE CHARACTER EXPANSION FACTOR                      |
| <code>ginq_char_ht</code>            | INQUIRE CHARACTER HEIGHT                                |
| <code>ginq_char_space</code>         | INQUIRE CHARACTER SPACING                               |
| <code>ginq_char_up_vec</code>        | INQUIRE CHARACTER UP VECTOR                             |
| <code>ginq_choice_st</code>          | INQUIRE CHOICE DEVICE STATE                             |
| <code>ginq_colr_facs</code>          | INQUIRE COLOUR FACILITIES                               |
| <code>ginq_colr_rep</code>           | INQUIRE COLOUR REPRESENTATION                           |
| <code>ginq_cur_cont_attrs</code>     | INQUIRE CURRENT CONTOUR ATTRIBUTES                      |
| <code>ginq_cur_norm_tran_num</code>  | INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER     |
| <code>ginq_cur_pick_id</code>        | INQUIRE CURRENT PICK IDENTIFIER VALUE                   |
| <code>ginq_def_choice_data</code>    | INQUIRE DEFAULT CHOICE DEVICE DATA                      |
| <code>ginq_def_font_ind_map</code>   | INQUIRE DEFAULT FONT INDEX MAPPING                      |
| <code>ginq_def_loc_data</code>       | INQUIRE DEFAULT LOCATOR DEVICE DATA                     |
| <code>ginq_def_pick_data</code>      | INQUIRE DEFAULT PICK DEVICE DATA                        |
| <code>ginq_def_string_data</code>    | INQUIRE DEFAULT STRING DEVICE DATA                      |
| <code>ginq_def_stroke_data</code>    | INQUIRE DEFAULT STROKE DEVICE DATA                      |
| <code>ginq_def_val_data</code>       | INQUIRE DEFAULT VALUATOR DEVICE DATA                    |
| <code>ginq_disp_space_size</code>    | INQUIRE DISPLAY SPACE SIZE                              |
| <code>ginq_edge_colr_ind</code>      | INQUIRE EDGE COLOUR INDEX                               |
| <code>ginq_edge_colr_specif</code>   | INQUIRE EDGE COLOUR SPECIFIER                           |
| <code>ginq_edge_flag</code>          | INQUIRE EDGE FLAG                                       |
| <code>ginq_edgetype</code>           | INQUIRE EDGETYPE                                        |
| <code>ginq_edgewidth</code>          | INQUIRE EDGEWIDTH SCALE FACTOR                          |
| <code>ginq_font_ind_map</code>       | INQUIRE FONT INDEX MAPPING                              |
| <code>ginq_gen_ws_dt_entry</code>    | INQUIRE GENERIC WORKSTATION DESCRIPTION TABLE ENTRY     |
| <code>ginq_gks_dt_entry</code>       | INQUIRE GKS DESCRIPTION TABLE ENTRY                     |
| <code>ginq_gks_sl_entry</code>       | INQUIRE GKS STATE LIST ENTRY                            |
| <code>ginq_global_tran_matrix</code> | INQUIRE GLOBAL TRANSFORMATION MATRIX                    |
| <code>ginq_in_dev_comps</code>       | INQUIRE INPUT DEVICE COMPOSITIONS                       |
| <code>ginq_in_dev_init_values</code> | INQUIRE INPUT DEVICE INITIAL VALUES                     |
| <code>ginq_in_overf</code>           | INQUIRE INPUT QUEUE OVERFLOW                            |
| <code>ginq_in_queue</code>           | INQUIRE INPUT QUEUE                                     |
| <code>ginq_in_table</code>           | INQUIRE INPUT DEVICE OPERATING MODES AND INITIAL VALUES |
| <code>ginq_int_colr_ind</code>       | INQUIRE INTERIOR COLOUR INDEX                           |
| <code>ginq_int_colr_specif</code>    | INQUIRE INTERIOR COLOUR SPECIFIER                       |
| <code>ginq_int_style</code>          | INQUIRE INTERIOR STYLE                                  |
| <code>ginq_int_style_ind</code>      | INQUIRE INTERIOR STYLE INDEX                            |
| <code>ginq_line_colr_ind</code>      | INQUIRE LINE COLOUR INDEX                               |
| <code>ginq_line_colr_specif</code>   | INQUIRE LINE COLOUR SPECIFIER                           |
| <code>ginq_line_facs</code>          | INQUIRE LINE FACILITIES                                 |
| <code>ginq_line_ind</code>           | INQUIRE LINE INDEX                                      |
| <code>ginq_line_rep</code>           | INQUIRE LINE REPRESENTATION                             |
| <code>ginq_linetype</code>           | INQUIRE LINETYPE                                        |
| <code>ginq_linewidth</code>          | INQUIRE LINEWIDTH SCALE FACTOR                          |
| <code>ginq_list_area_inds</code>     | INQUIRE LIST OF AREA INDICES                            |
| <code>ginq_list_avail_fonts</code>   | INQUIRE LIST OF AVAILABLE FONTS                         |

|                          |                                                          |
|--------------------------|----------------------------------------------------------|
| ginq_list_avail_gdps     | INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES |
| ginq_list_avail_ws_types | INQUIRE LIST OF AVAILABLE WORKSTATION TYPES              |
| ginq_list_colr_inds      | INQUIRE LIST OF COLOUR INDICES                           |
| ginq_list_line_inds      | INQUIRE LIST OF LINE INDICES                             |
| ginq_list_marker_inds    | INQUIRE LIST OF MARKER INDICES                           |
| ginq_list_norm_trans     | INQUIRE LIST OF NORMALIZATION TRANSFORMATIONS            |
| ginq_list_pat_inds       | INQUIRE LIST OF PATTERN INDICES                          |
| ginq_list_text_inds      | INQUIRE LIST OF TEXT INDICES                             |
| ginq_list_vp_in_pris     | INQUIRE LIST OF VIEWPORT INPUT PRIORITIES                |
| ginq_loc_st              | INQUIRE LOCATOR DEVICE STATE                             |
| ginq_local_tran_matrix   | INQUIRE LOCAL TRANSFORMATION MATRIX                      |
| ginq_marker_colr_ind     | INQUIRE MARKER COLOUR INDEX                              |
| ginq_marker_colr_specif  | INQUIRE MARKER COLOUR SPECIFIER                          |
| ginq_marker_facs         | INQUIRE MARKER FACILITIES                                |
| ginq_marker_ind          | INQUIRE MARKER INDEX                                     |
| ginq_marker_rep          | INQUIRE MARKER REPRESENTATION                            |
| ginq_marker_size         | INQUIRE MARKER SIZE SCALE FACTOR                         |
| ginq_marker_type         | INQUIRE MARKER TYPE                                      |
| ginq_name_open_pic_part  | INQUIRE NAME OF OPEN PICTURE PART                        |
| ginq_name_open_seg       | INQUIRE NAME OF OPEN SEGMENT                             |
| ginq_name_open_stencil   | INQUIRE NAME OF OPEN STENCIL                             |
| ginq_name_open_tiling    | INQUIRE NAME OF OPEN TILING                              |
| ginq_nameset             | INQUIRE NAMESSET                                         |
| ginq_norm_tran           | INQUIRE NORMALIZATION TRANSFORMATION                     |
| ginq_num_avail_in        | INQUIRE NUMBER OF AVAILABLE INPUT DEVICES                |
| ginq_op_st_entry         | INQUIRE OPERATING STATE ENTRY                            |
| ginq_pat_facs            | INQUIRE PATTERN FACILITIES                               |
| ginq_pat_ref_point       | INQUIRE PATTERN REFERENCE POINT                          |
| ginq_pat_rep             | INQUIRE PATTERN REPRESENTATION                           |
| ginq_pat_size            | INQUIRE PATTERN SIZE                                     |
| ginq_pick_st             | INQUIRE PICK DEVICE STATE                                |
| ginq_pred_area_rep       | INQUIRE PREDEFINED AREA REPRESENTATION                   |
| ginq_pred_colr_rep       | INQUIRE PREDEFINED COLOUR REPRESENTATION                 |
| ginq_pred_line_rep       | INQUIRE PREDEFINED LINE REPRESENTATION                   |
| ginq_pred_marker_rep     | INQUIRE PREDEFINED MARKER REPRESENTATION                 |
| ginq_pred_pat_rep        | INQUIRE PREDEFINED PATTERN REPRESENTATION                |
| ginq_pred_text_rep       | INQUIRE PREDEFINED TEXT REPRESENTATION                   |
| ginq_route_dir           | INQUIRE ROUTE DIRECTION                                  |
| ginq_scissor_mode        | INQUIRE SCISSOR MODE                                     |
| ginq_scissor_set         | INQUIRE SCISSOR SET                                      |
| ginq_seg_attrs           | INQUIRE SEGMENT ATTRIBUTES                               |
| ginq_sel_table           | INQUIRE TABLE OF SELECTION CRITERIA                      |
| ginq_set_active_wss      | INQUIRE SET OF ACTIVE WORKSTATIONS                       |
| ginq_set_assoc_wss       | INQUIRE SET OF ASSOCIATED WORKSTATIONS                   |
| ginq_set_avail_in_devs   | INQUIRE SET OF AVAILABLE INPUT DEVICES                   |
| ginq_set_avail_meass     | INQUIRE SET OF AVAILABLE MEASURES                        |
| ginq_set_avail_triggers  | INQUIRE SET OF AVAILABLE TRIGGERS                        |
| ginq_set_open_audits     | INQUIRE SET OF OPEN AUDITS                               |
| ginq_set_open_playbacks  | INQUIRE SET OF OPEN PLAYBACKS                            |
| ginq_set_open_wss        | INQUIRE SET OF OPEN WORKSTATIONS                         |
| ginq_set_pic_part_names  | INQUIRE SET OF PICTURE PART NAMES IN USE                 |
| ginq_set_seg_names       | INQUIRE SET OF SEGMENT NAMES IN USE                      |
| ginq_set_seg_names_ws    | INQUIRE SET OF SEGMENT NAMES ON WORKSTATION              |
| ginq_set_stencil_names   | INQUIRE SET OF STENCIL NAMES IN USE                      |
| ginq_set_tiling_names    | INQUIRE SET OF TILING NAMES IN USE                       |

|                                         |                                                            |
|-----------------------------------------|------------------------------------------------------------|
| <b>ginq_specif_ws_dt_entry</b>          | INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY       |
| <b>ginq_specif_ws_dt_entry_st</b>       | INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY STATE |
| <b>ginq_stencil_attr</b>                | INQUIRE STENCIL ATTRIBUTES                                 |
| <b>ginq_string_st</b>                   | INQUIRE STRING DEVICE STATE                                |
| <b>ginq_stroke_st</b>                   | INQUIRE STROKE DEVICE STATE                                |
| <b>ginq_text_align</b>                  | INQUIRE TEXT ALIGNMENT                                     |
| <b>ginq_text_colr_ind</b>               | INQUIRE TEXT COLOUR INDEX                                  |
| <b>ginq_text_colr_specif</b>            | INQUIRE TEXT COLOUR SPECIFIER                              |
| <b>ginq_text_extent</b>                 | INQUIRE TEXT EXTENT                                        |
| <b>ginq_text_fac</b>                    | INQUIRE TEXT FACILITIES                                    |
| <b>ginq_text_font_prec</b>              | INQUIRE TEXT FONT AND PRECISION                            |
| <b>ginq_text_ind</b>                    | INQUIRE TEXT INDEX                                         |
| <b>ginq_text_path</b>                   | INQUIRE TEXT PATH                                          |
| <b>ginq_text_rep</b>                    | INQUIRE TEXT REPRESENTATION                                |
| <b>ginq_text_skew_angle</b>             | INQUIRE TEXT SKEW ANGLE                                    |
| <b>ginq_val_st</b>                      | INQUIRE VALUATOR DEVICE STATE                              |
| <b>ginq_view_pris</b>                   | INQUIRE VIEW PRIORITIES                                    |
| <b>ginq_view_rep</b>                    | INQUIRE VIEW REPRESENTATION                                |
| <b>ginq_vis_effects_st</b>              | INQUIRE VISUAL EFFECTS STATE                               |
| <b>ginq_ws_class</b>                    | INQUIRE WORKSTATION CLASSIFICATION                         |
| <b>ginq_ws_conn_type</b>                | INQUIRE WORKSTATION CONNECTION AND TYPE                    |
| <b>ginq_ws_sl_entry</b>                 | INQUIRE WORKSTATION STATE LIST ENTRY                       |
| <b>ginq_ws_st</b>                       | INQUIRE WORKSTATION STATE                                  |
| <b>ginq_ws_win_vp</b>                   | INQUIRE WORKSTATION WINDOW AND VIEWPORT                    |
| <b>ginsert_seg</b>                      | INSERT SEGMENT                                             |
| <b>ginst_stencil</b>                    | INSTANCE STENCIL                                           |
| <b>ginst_stencil_path</b>               | INSTANCE STENCIL ALONG PATH                                |
| <b>ginst_stencil_seq_path</b>           | INSTANCE STENCIL SEQUENCE ALONG PATH                       |
| <b>gmessage</b>                         | MESSAGE                                                    |
| <b>gnurb_set</b>                        | NURB SET                                                   |
| <b>gopen_gks</b>                        | OPEN GKS                                                   |
| <b>gopen_pic_part</b>                   | OPEN PICTURE PART                                          |
| <b>gopen_stencil</b>                    | OPEN STENCIL                                               |
| <b>gopen_tiling</b>                     | OPEN TILING                                                |
| <b>gopen_ws</b>                         | OPEN WORKSTATION                                           |
| <b>gplayback</b>                        | PLAYBACK                                                   |
| <b>gpolyline</b>                        | POLYLINE                                                   |
| <b>gpolyline_set</b>                    | POLYLINE SET                                               |
| <b>gpolymarker</b>                      | POLYMARKER                                                 |
| <b>gproc_audit_item</b>                 | PROCESS AUDIT ITEM                                         |
| <b>gread_item_audit</b>                 | READ ITEM FROM AUDIT                                       |
| <b>gread_item_func_name_audit</b>       | READ ITEM FUNCTION NAME FROM AUDIT                         |
| <b>gremove_backdrop</b>                 | REMOVE BACKDROP                                            |
| <b>gremove_set_names_nameset</b>        | REMOVE SET OF NAMES FROM NAMESET                           |
| <b>gremove_set_names_ndc_pic</b>        | REMOVE SET OF NAMES FROM NDC PICTURE                       |
| <b>gremove_set_scissors_ndc_pic</b>     | REMOVE SET OF SCISSORS FROM NDC PICTURE                    |
| <b>gremove_set_scissors_scissor_set</b> | REMOVE SET OF SCISSORS FROM SCISSOR SET                    |
| <b>grename_pic_part</b>                 | RENAME PICTURE PART                                        |
| <b>grename_seg</b>                      | RENAME SEGMENT                                             |
| <b>grename_stencil</b>                  | RENAME STENCIL                                             |
| <b>grename_tiling</b>                   | RENAME TILING                                              |
| <b>greopen_pic_part</b>                 | REOPEN PICTURE PART                                        |
| <b>greorder_ndc_pic</b>                 | REORDER NDC PICTURE                                        |
| <b>greq_choice</b>                      | REQUEST CHOICE                                             |

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| greq_in                      | REQUEST INPUT                                            |
| greq_loc                     | REQUEST LOCATOR                                          |
| greq_pick                    | REQUEST PICK                                             |
| greq_string                  | REQUEST STRING                                           |
| greq_stroke                  | REQUEST STROKE                                           |
| greq_val                     | REQUEST VALUATOR                                         |
| greset_specif_ws_dt_entry_st | RESET SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY STATE |
| grestore_gks_sl              | RESTORE GKS STATE LIST                                   |
| grestore_ws_sl               | RESTORE WORKSTATION STATE LIST                           |
| gret_pic_part_ar             | RETRIEVE PICTURE PART FROM ARCHIVE                       |
| groute                       | ROUTE                                                    |
| gsample_choice               | SAMPLE CHOICE                                            |
| gsample_in                   | SAMPLE INPUT                                             |
| gsample_loc                  | SAMPLE LOCATOR                                           |
| gsample_pick                 | SAMPLE PICK                                              |
| gsample_string               | SAMPLE STRING                                            |
| gsample_stroke               | SAMPLE STROKE                                            |
| gsample_val                  | SAMPLE VALUATOR                                          |
| gsave_gks_sl                 | SAVE GKS STATE LIST                                      |
| gsave_ws_sl                  | SAVE WORKSTATION STATE LIST                              |
| gsel_norm_tran               | SELECT NORMALIZATION TRANSFORMATION                      |
| gset_area_ind                | SET AREA INDEX                                           |
| gset_area_rep                | SET AREA REPRESENTATION                                  |
| gset_asfs                    | SET ATTRIBUTE SOURCE FLAGS                               |
| gset_char_code               | SET CHARACTER CODE                                       |
| 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_colr_rep                | SET COLOUR REPRESENTATION                                |
| gset_cont_attr               | SET CONTOUR ATTRIBUTE                                    |
| gset_det                     | SET DETECTABILITY                                        |
| gset_edge_colr_ind           | SET EDGE COLOUR INDEX                                    |
| gset_edge_colr_specif        | SET EDGE COLOUR SPECIFIER                                |
| gset_edge_flag               | SET EDGE FLAG                                            |
| gset_edgetype                | SET EDGETYPE                                             |
| gset_edgewidth               | SET EDGEWIDTH SCALE FACTOR                               |
| gset_err_hand                | SET ERROR HANDLING                                       |
| gset_font_ind_map            | SET FONT INDEX MAPPING                                   |
| gset_global_tran_matrix      | SET GLOBAL TRANSFORMATION MATRIX                         |
| gset_highl                   | SET HIGHLIGHTING                                         |
| gset_in_dev_mode             | SET LOGICAL INPUT DEVICE MODE                            |
| gset_int_colr_ind            | SET INTERIOR COLOUR INDEX                                |
| gset_int_colr_specif         | SET INTERIOR COLOUR SPECIFIER                            |
| gset_int_style               | SET INTERIOR STYLE                                       |
| gset_int_style_ind           | SET INTERIOR STYLE INDEX                                 |
| gset_line_colr_ind           | SET LINE COLOUR INDEX                                    |
| gset_line_colr_specif        | SET LINE COLOUR SPECIFIER                                |
| gset_line_ind                | SET LINE INDEX                                           |
| gset_line_rep                | SET LINE REPRESENTATION                                  |
| gset_linetype                | SET LINETYPE                                             |
| gset_linewidth               | SET LINEWIDTH SCALE FACTOR                               |
| gset_loc_mode                | SET LOCATOR MODE                                         |
| gset_local_tran_matrix       | SET LOCAL TRANSFORMATION MATRIX                          |

|                                      |                                         |
|--------------------------------------|-----------------------------------------|
| <code>gset_marker_colr_ind</code>    | SET LINE COLOUR INDEX                   |
| <code>gset_marker_colr_specif</code> | SET MARKER COLOUR SPECIFIER             |
| <code>gset_marker_ind</code>         | SET MARKER INDEX                        |
| <code>gset_marker_rep</code>         | SET MARKER REPRESENTATION               |
| <code>gset_marker_size</code>        | SET MARKER SIZE SCALE FACTOR            |
| <code>gset_marker_type</code>        | SET MARKER TYPE                         |
| <code>gset_nameset</code>            | SET NAMESET                             |
| <code>gset_ndc_pic_prim_attr</code>  | SET NDC PICTURE PRIMITIVE ATTRIBUTE     |
| <code>gset_norm_tran_num</code>      | SET NORMALIZATION TRANSFORMATION NUMBER |
| <code>gset_pat_ref_point</code>      | SET PATTERN REFERENCE POINT             |
| <code>gset_pat_rep</code>            | SET PATTERN REPRESENTATION              |
| <code>gset_pat_size</code>           | SET PATTERN SIZE                        |
| <code>gset_pick_id</code>            | SET PICK IDENTIFIER                     |
| <code>gset_pick_mode</code>          | SET PICK MODE                           |
| <code>gset_prim_attr</code>          | SET PRIMITIVE ATTRIBUTE                 |
| <code>gset_rep</code>                | SET REPRESENTATION                      |
| <code>gset_scissor_mode</code>       | SET SCISSOR MODE                        |
| <code>gset_scissor_set</code>        | SET SCISSOR SET                         |
| <code>gset_seg_attr</code>           | SET SEGMENT ATTRIBUTE                   |
| <code>gset_seg_pri</code>            | SET SEGMENT PRIORITY                    |
| <code>gset_seg_tran</code>           | SET SEGMENT TRANSFORMATION              |
| <code>gset_stencil_attr</code>       | SET STENCIL ATTRIBUTE                   |
| <code>gset_string_mode</code>        | SET STRING MODE                         |
| <code>gset_stroke_mode</code>        | SET STROKE MODE                         |
| <code>gset_text_align</code>         | SET TEXT ALIGNMENT                      |
| <code>gset_text_colr_ind</code>      | SET TEXT COLOUR INDEX                   |
| <code>gset_text_colr_specif</code>   | SET TEXT COLOUR SPECIFIER               |
| <code>gset_text_font_prec</code>     | SET TEXT FONT AND PRECISION             |
| <code>gset_text_ind</code>           | SET TEXT INDEX                          |
| <code>gset_text_path</code>          | SET TEXT PATH                           |
| <code>gset_text_rep</code>           | SET TEXT REPRESENTATION                 |
| <code>gset_text_skew_angle</code>    | SET TEXT SKEW ANGLE                     |
| <code>gset_val_mode</code>           | SET VALUATOR MODE                       |
| <code>gset_view</code>               | SET VIEW                                |
| <code>gset_view_pri</code>           | SET VIEW PRIORITY                       |
| <code>gset_view_sel_crit</code>      | SET VIEW SELECTION CRITERION            |
| <code>gset_vis</code>                | SET VISIBILITY                          |
| <code>gset_vp</code>                 | SET VIEWPORT                            |
| <code>gset_vp_in_pri</code>          | SET VIEWPORT INPUT PRIORITY             |
| <code>gset_win</code>                | SET WINDOW                              |
| <code>gset_win_vp</code>             | SET WINDOW AND VIEWPORT                 |
| <code>gset_ws_sel_crit</code>        | SET WORKSTATION SELECTION CRITERION     |
| <code>gset_ws_vis_effects</code>     | SET WORKSTATION VISUAL EFFECTS          |
| <code>gset_ws_vp</code>              | SET WORKSTATION VIEWPORT                |
| <code>gset_ws_win</code>             | SET WORKSTATION WINDOW                  |
| <code>gset_ws_win_vp</code>          | SET WORKSTATION WINDOW AND VIEWPORT     |
| <code>gtext</code>                   | TEXT                                    |
| <code>gwrite_user_rec_audit</code>   | WRITE USER RECORD TO AUDIT              |

#### 4.3.2 List ordered alphabetically by GKS name

|                                    |                                        |
|------------------------------------|----------------------------------------|
| ACCUMULATE TRANSFORMATION MATRIX   | <code>gaccum_tran_matrix</code>        |
| ACTIVATE WORKSTATION               | <code>gactivate_ws</code>              |
| ADD SET OF NAMES TO NAMESET        | <code>gadd_set_names_nameset</code>    |
| ADD SET OF NAMES TO NDC PICTURE    | <code>gadd_set_names_ndc_pic</code>    |
| ADD SET OF SCISSORS TO NDC PICTURE | <code>gadd_set_scissors_ndc_pic</code> |

|                                                  |                               |
|--------------------------------------------------|-------------------------------|
| ADD SET OF SCISSORS TO SCISSOR SET               | gadd_set_scissors_scissor_set |
| APPEND PICTURE PART                              | gappend_pic_part              |
| ARCHIVE PICTURE PART                             | gar_pic_part                  |
| ASSOCIATE SEGMENT WITH WORKSTATION               | gassoc_seg_ws                 |
| AUDIT                                            | gaudit                        |
| AWAIT EVENT                                      | gawait_event                  |
| AWAIT INPUT                                      | gawait_in                     |
| CELL ARRAY                                       | gcell_array                   |
| CLEAR WORKSTATION                                | gclear_ws                     |
| CLOSE GKS                                        | gclose_gks                    |
| CLOSE PICTURE PART                               | gclose_pic_part               |
| CLOSE SEGMENT                                    | gclose_seg                    |
| CLOSE STENCIL                                    | gclose_stencil                |
| CLOSE TILING                                     | gclose_tiling                 |
| CLOSE WORKSTATION                                | gclose_ws                     |
| CLOSED NURB SET                                  | gclosed_nurb_set              |
| CONIC SECTION SET                                | gconic_sec_set                |
| CONVERT COLOUR                                   | gconv_colr                    |
| COPY BLANK REALIZED PICTURE TO REALIZED METAFILE | gcopy_blank_real_pic_real_mf  |
| COPY NDC METAFILE PICTURE TO NDC PICTURE         | gcopy_ndc_mf_pic_ndc_pic      |
| COPY NDC PICTURE TO NDC METAFILE                 | gcopy_ndc_pic_ndc_mf          |
| COPY NDC PICTURE TO PICTURE PART STORE           | gcopy_ndc_pic_pic_part_store  |
| COPY PICTURE PART FROM PICTURE PART STORE        | gcopy_pic_part_pic_part_store |
| COPY REALIZED METAFILE PICTURE TO BACKDROP       | gcopy_real_mf_pic_backdrop    |
| COPY REALIZED PICTURE TO REALIZED METAFILE       | gcopy_real_pic_real_mf        |
| COPY SEGMENT FROM WORKSTATION                    | gcopy_seg_ws                  |
| CREATE OUTPUT PRIMITIVE                          | gcreate_out_prim              |
| CREATE SEGMENT                                   | gcreate_seg                   |
| CREATE STENCIL FROM BOUNDARY                     | gcreate_stencil_bndry         |
| CREATE STENCIL FROM CONTOURS                     | gcreate_stencil_conts         |
| CREATE STORAGE                                   | gcreate_store                 |
| CREATE TILING COMPONENT                          | gcreate_tiling_comp           |
| DEACTIVATE WORKSTATION                           | gdeactivate_ws                |
| DEFINE LOGICAL INPUT DEVICE                      | gdef_in_dev                   |
| DELETE PICTURE PART                              | gdel_pic_part                 |
| DELETE PRIMITIVES FROM NDC PICTURE               | gdel_prims_ndc_pic            |
| DELETE SEGMENT                                   | gdel_seg                      |
| DELETE SEGMENT FROM WORKSTATION                  | gdel_seg_ws                   |
| DELETE STENCIL                                   | gdel_stencil                  |
| DELETE STORAGE                                   | gdel_store                    |
| DELETE TILING                                    | gdel_tiling                   |
| DESIGN                                           | gdesign                       |
| ELLIPTIC DISC SET                                | gell_disc_set                 |
| ELLIPTIC SECTOR SET                              | gell_sec_set                  |
| ELLIPTIC SEGMENT SET                             | gell_seg_set                  |
| EMERGENCY CLOSE GKS                              | gemergency_close_gks          |
| ERROR HANDLING                                   | gerr_hand                     |
| ERROR LOGGING                                    | gerr_log                      |
| ESCAPE                                           | gescape                       |
| EVALUATE CIRCLE                                  | geval_circle                  |
| EVALUATE CIRCULAR ARC 3 POINT                    | geval_circ_arc_3_point        |
| EVALUATE CIRCULAR ARC CENTRE                     | geval_circ_arc_ctr            |
| EVALUATE ELLIPSE                                 | geval_ell                     |
| EVALUATE ELLIPTIC ARC                            | geval_ell_arc                 |
| EVALUATE HYPERBOLIC ARC                          | geval_hyp_arc                 |

|                                                     |                        |
|-----------------------------------------------------|------------------------|
| EVALUATE PARABOLIC ARC                              | geval_par_arc          |
| EVALUATE TRANSFORMATION MATRIX                      | geval_tran_matrix      |
| EVALUATE VIEW MAPPING MATRIX                        | geval_view_map_matrix  |
| EVALUATE VIEW ORIENTATION MATRIX                    | geval_view_ori_matrix  |
| EVALUATE WC ORDINATE                                | geval_wc_ord           |
| FILL AREA                                           | gfill_area             |
| FILL AREA SET                                       | gfill_area_set         |
| FLUSH DEVICE EVENTS                                 | gflush_dev_events      |
| GENERALIZED DRAWING PRIMITIVE                       | ggdp                   |
| GET CHOICE                                          | gget_choice            |
| GET GLYPH NAME                                      | gget_glyph_name        |
| GET LOCATOR                                         | gget_loc               |
| GET MAPPED SEGMENT NAME                             | gget_map_seg_name      |
| GET MAPPED WORKSTATION IDENTIFIER                   | gget_map_ws_id         |
| GET PICK                                            | gget_pick              |
| GET STENCIL ATTRIBUTE                               | gget_stencil_attr      |
| GET STRING                                          | gget_string            |
| GET STROKE                                          | gget_stroke            |
| GET TEXT EXTENT                                     | gget_text_extent       |
| GET VALUATOR                                        | gget_val               |
| GET WORKSTATION STATUS                              | gget_ws_status         |
| INITIALIZE CHOICE                                   | ginit_choice           |
| INITIALIZE LOCATOR                                  | ginit_loc              |
| INITIALIZE LOGICAL INPUT DEVICE                     | ginit_in_dev           |
| INITIALIZE PICK                                     | ginit_pick             |
| INITIALIZE STRING                                   | ginit_string           |
| INITIALIZE STROKE                                   | ginit_stroke           |
| INITIALIZE VALUATOR                                 | ginit_val              |
| INQUIRE AREA FACILITIES                             | ginq_area_fac          |
| INQUIRE AREA INDEX                                  | ginq_area_ind          |
| INQUIRE AREA REPRESENTATION                         | ginq_area_rep          |
| INQUIRE ATTRIBUTE SOURCE FLAGS                      | ginq_asfs              |
| 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 CHOICE DEVICE STATE                         | ginq_choice_st         |
| INQUIRE COLOUR FACILITIES                           | ginq_colr_fac          |
| INQUIRE COLOUR REPRESENTATION                       | ginq_colr_rep          |
| INQUIRE CURRENT CONTOUR ATTRIBUTES                  | ginq_cur_cont_attrs    |
| INQUIRE CURRENT NORMALIZATION TRANSFORMATION NUMBER | ginq_cur_norm_tran_num |
| INQUIRE CURRENT PICK IDENTIFIER VALUE               | ginq_cur_pick_id       |
| INQUIRE DEFAULT CHOICE DEVICE DATA                  | ginq_def_choice_data   |
| INQUIRE DEFAULT FONT INDEX MAPPING                  | ginq_def_font_ind_map  |
| INQUIRE DEFAULT LOCATOR DEVICE DATA                 | ginq_def_loc_data      |
| INQUIRE DEFAULT PICK DEVICE DATA                    | ginq_def_pick_data     |
| INQUIRE DEFAULT STRING DEVICE DATA                  | ginq_def_string_data   |
| INQUIRE DEFAULT STROKE DEVICE DATA                  | ginq_def_stroke_data   |
| INQUIRE DEFAULT VALUATOR DEVICE DATA                | ginq_def_val_data      |
| INQUIRE DISPLAY SPACE SIZE                          | ginq_disp_space_size   |
| INQUIRE EDGE COLOUR INDEX                           | ginq_edge_colr_ind     |
| INQUIRE EDGE COLOUR SPECIFIER                       | ginq_edge_colr_specif  |
| INQUIRE EDGE FLAG                                   | ginq_edge_flag         |
| INQUIRE EDGETYPE                                    | ginq_edgetype          |
| INQUIRE EDGEWIDTH SCALE FACTOR                      | ginq_edgewidth         |

|                                                          |                          |
|----------------------------------------------------------|--------------------------|
| INQUIRE FONT INDEX MAPPING                               | ginq_font_ind_map        |
| INQUIRE GENERIC WORKSTATION DESCRIPTION TABLE ENTRY      | ginq_gen_ws_dt_entry     |
| INQUIRE GKS DESCRIPTION TABLE ENTRY                      | ginq_gks_dt_entry        |
| INQUIRE GKS STATE LIST ENTRY                             | ginq_gks_sl_entry        |
| INQUIRE GLOBAL TRANSFORMATION MATRIX                     | ginq_global_tran_matrix  |
| INQUIRE INPUT DEVICE COMPOSITIONS                        | ginq_in_dev_comps        |
| INQUIRE INPUT DEVICE INITIAL VALUES                      | ginq_in_dev_init_values  |
| INQUIRE INPUT DEVICE OPERATING MODES AND INITIAL VALUES  | ginq_in_table            |
| INQUIRE INPUT QUEUE                                      | ginq_in_queue            |
| INQUIRE INPUT QUEUE OVERFLOW                             | ginq_in_overf            |
| INQUIRE INTERIOR COLOUR INDEX                            | ginq_int_colr_ind        |
| INQUIRE INTERIOR COLOUR SPECIFIER                        | ginq_int_colr_specif     |
| INQUIRE INTERIOR STYLE                                   | ginq_int_style           |
| INQUIRE INTERIOR STYLE INDEX                             | ginq_int_style_ind       |
| INQUIRE LINE COLOUR INDEX                                | ginq_line_colr_ind       |
| INQUIRE LINE COLOUR SPECIFIER                            | ginq_line_colr_specif    |
| INQUIRE LINE FACILITIES                                  | ginq_line_facs           |
| INQUIRE LINE INDEX                                       | ginq_line_ind            |
| INQUIRE LINE REPRESENTATION                              | ginq_line_rep            |
| INQUIRE LINETYPE                                         | ginq_linetype            |
| INQUIRE LINEWIDTH SCALE FACTOR                           | ginq_linewidth           |
| INQUIRE LIST OF AREA INDICES                             | ginq_list_area_inds      |
| INQUIRE LIST OF AVAILABLE FONTS                          | ginq_list_avail_fonts    |
| INQUIRE LIST OF AVAILABLE GENERALIZED DRAWING PRIMITIVES | ginq_list_avail_gdps     |
| INQUIRE LIST OF AVAILABLE WORKSTATION TYPES              | ginq_list_avail_ws_types |
| INQUIRE LIST OF COLOUR INDICES                           | ginq_list_colr_inds      |
| INQUIRE LIST OF LINE INDICES                             | ginq_list_line_inds      |
| INQUIRE LIST OF MARKER INDICES                           | ginq_list_marker_inds    |
| INQUIRE LIST OF NORMALIZATION TRANSFORMATIONS            | ginq_list_norm_trans     |
| INQUIRE LIST OF PATTERN INDICES                          | ginq_list_pat_inds       |
| INQUIRE LIST OF TEXT INDICES                             | ginq_list_text_inds      |
| INQUIRE LIST OF VIEWPORT INPUT PRIORITIES                | ginq_list_vp_in_pris     |
| INQUIRE LOCAL TRANSFORMATION MATRIX                      | ginq_local_tran_matrix   |
| INQUIRE LOCATOR DEVICE STATE                             | ginq_loc_st              |
| INQUIRE MARKER COLOUR INDEX                              | ginq_marker_colr_ind     |
| INQUIRE MARKER COLOUR SPECIFIER                          | ginq_marker_colr_specif  |
| INQUIRE MARKER FACILITIES                                | ginq_marker_facs         |
| INQUIRE MARKER INDEX                                     | ginq_marker_ind          |
| INQUIRE MARKER REPRESENTATION                            | ginq_marker_rep          |
| INQUIRE MARKER SIZE SCALE FACTOR                         | ginq_marker_size         |
| INQUIRE MARKER TYPE                                      | ginq_marker_type         |
| INQUIRE NAME OF OPEN PICTURE PART                        | ginq_name_open_pic_part  |
| INQUIRE NAME OF OPEN SEGMENT                             | ginq_name_open_seg       |
| INQUIRE NAME OF OPEN STENCIL                             | ginq_name_open_stencil   |
| INQUIRE NAME OF OPEN TILING                              | ginq_name_open_tiling    |
| INQUIRE NAMESET                                          | ginq_nameset             |
| INQUIRE NORMALIZATION TRANSFORMATION                     | ginq_norm_tran           |
| INQUIRE NUMBER OF AVAILABLE INPUT DEVICES                | ginq_num_avail_in        |
| INQUIRE OPERATING STATE ENTRY                            | ginq_op_st_entry         |
| INQUIRE PATTERN FACILITIES                               | ginq_pat_facs            |
| INQUIRE PATTERN REFERENCE POINT                          | ginq_pat_ref_point       |
| INQUIRE PATTERN REPRESENTATION                           | ginq_pat_rep             |
| INQUIRE PATTERN SIZE                                     | ginq_pat_size            |
| INQUIRE PICK DEVICE STATE                                | ginq_pick_st             |
| INQUIRE PREDEFINED AREA REPRESENTATION                   | ginq_pred_area_rep       |

|                                                            |                                         |
|------------------------------------------------------------|-----------------------------------------|
| INQUIRE PREDEFINED COLOUR REPRESENTATION                   | <code>ginq_pred_colr_rep</code>         |
| INQUIRE PREDEFINED LINE REPRESENTATION                     | <code>ginq_pred_line_rep</code>         |
| INQUIRE PREDEFINED MARKER REPRESENTATION                   | <code>ginq_pred_marker_rep</code>       |
| INQUIRE PREDEFINED PATTERN REPRESENTATION                  | <code>ginq_pred_pat_rep</code>          |
| INQUIRE PREDEFINED TEXT REPRESENTATION                     | <code>ginq_pred_text_rep</code>         |
| INQUIRE ROUTE DIRECTION                                    | <code>ginq_route_dir</code>             |
| INQUIRE SCISSOR MODE                                       | <code>ginq_scissor_mode</code>          |
| INQUIRE SCISSOR SET                                        | <code>ginq_scissor_set</code>           |
| INQUIRE SEGMENT ATTRIBUTES                                 | <code>ginq_seg_attrs</code>             |
| INQUIRE SET OF ACTIVE WORKSTATIONS                         | <code>ginq_set_active_wss</code>        |
| INQUIRE SET OF ASSOCIATED WORKSTATIONS                     | <code>ginq_set_assoc_wss</code>         |
| INQUIRE SET OF AVAILABLE INPUT DEVICES                     | <code>ginq_set_avail_in_devs</code>     |
| INQUIRE SET OF AVAILABLE MEASURES                          | <code>ginq_set_avail_meass</code>       |
| INQUIRE SET OF AVAILABLE TRIGGERS                          | <code>ginq_set_avail_triggers</code>    |
| INQUIRE SET OF OPEN AUDITS                                 | <code>ginq_set_open_audits</code>       |
| INQUIRE SET OF OPEN PLAYBACKS                              | <code>ginq_set_open_playbacks</code>    |
| INQUIRE SET OF OPEN WORKSTATIONS                           | <code>ginq_set_open_wss</code>          |
| INQUIRE SET OF PICTURE PART NAMES IN USE                   | <code>ginq_set_pic_part_names</code>    |
| INQUIRE SET OF SEGMENT NAMES IN USE                        | <code>ginq_set_seg_names</code>         |
| INQUIRE SET OF SEGMENT NAMES ON WORKSTATION                | <code>ginq_set_seg_names_ws</code>      |
| INQUIRE SET OF STENCIL NAMES IN USE                        | <code>ginq_set_stencil_names</code>     |
| INQUIRE SET OF TILING NAMES IN USE                         | <code>ginq_set_tiling_names</code>      |
| INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY       | <code>ginq_specif_ws_dt_entry</code>    |
| INQUIRE SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY STATE | <code>ginq_specif_ws_dt_entry_st</code> |
| INQUIRE STENCIL ATTRIBUTES                                 | <code>ginq_stencil_attrs</code>         |
| INQUIRE STRING DEVICE STATE                                | <code>ginq_string_st</code>             |
| INQUIRE STROKE DEVICE STATE                                | <code>ginq_stroke_st</code>             |
| INQUIRE TABLE OF SELECTION CRITERIA                        | <code>ginq_sel_table</code>             |
| INQUIRE TEXT ALIGNMENT                                     | <code>ginq_text_align</code>            |
| INQUIRE TEXT COLOUR INDEX                                  | <code>ginq_text_colr_ind</code>         |
| INQUIRE TEXT COLOUR SPECIFIER                              | <code>ginq_text_colr_specif</code>      |
| INQUIRE TEXT EXTENT                                        | <code>ginq_text_extent</code>           |
| INQUIRE TEXT FACILITIES                                    | <code>ginq_text_fac</code>              |
| INQUIRE TEXT FONT AND PRECISION                            | <code>ginq_text_font_prec</code>        |
| INQUIRE TEXT INDEX                                         | <code>ginq_text_ind</code>              |
| INQUIRE TEXT PATH                                          | <code>ginq_text_path</code>             |
| INQUIRE TEXT REPRESENTATION                                | <code>ginq_text_rep</code>              |
| INQUIRE TEXT SKEW ANGLE                                    | <code>ginq_text_skew_angle</code>       |
| INQUIRE VALUATOR DEVICE STATE                              | <code>ginq_val_st</code>                |
| INQUIRE VIEW PRIORITIES                                    | <code>ginq_view_pris</code>             |
| INQUIRE VIEW REPRESENTATION                                | <code>ginq_view_rep</code>              |
| INQUIRE VISUAL EFFECTS STATE                               | <code>ginq_vis_effects_st</code>        |
| INQUIRE WORKSTATION CLASSIFICATION                         | <code>ginq_ws_class</code>              |
| INQUIRE WORKSTATION CONNECTION AND TYPE                    | <code>ginq_ws_conn_type</code>          |
| INQUIRE WORKSTATION STATE                                  | <code>ginq_ws_st</code>                 |
| INQUIRE WORKSTATION STATE LIST ENTRY                       | <code>ginq_ws_sl_entry</code>           |
| INQUIRE WORKSTATION WINDOW AND VIEWPORT                    | <code>ginq_ws_win_vp</code>             |
| INSERT SEGMENT                                             | <code>ginsert_seg</code>                |
| INSTANCE STENCIL                                           | <code>ginst_stencil</code>              |
| INSTANCE STENCIL ALONG PATH                                | <code>ginst_stencil_path</code>         |
| INSTANCE STENCIL SEQUENCE ALONG PATH                       | <code>ginst_stencil_seq_path</code>     |
| MESSAGE                                                    | <code>gmessage</code>                   |
| NURB SET                                                   | <code>gnurb_set</code>                  |
| OPEN GKS                                                   | <code>gopen_gks</code>                  |
| OPEN PICTURE PART                                          | <code>gopen_pic_part</code>             |

|                                                          |                                  |
|----------------------------------------------------------|----------------------------------|
| OPEN STENCIL                                             | gopen_stencil                    |
| OPEN TILING                                              | gopen_tiling                     |
| OPEN WORKSTATION                                         | gopen_ws                         |
| PLAYBACK                                                 | gplayback                        |
| POLYLINE                                                 | gpolyline                        |
| POLYLINE SET                                             | gpolyline_set                    |
| POLYMARKER                                               | gpolymarker                      |
| PROCESS AUDIT ITEM                                       | gproc_audit_item                 |
| READ ITEM FROM AUDIT                                     | gread_item_audit                 |
| READ ITEM FUNCTION NAME FROM AUDIT                       | gread_item_func_name_audit       |
| REMOVE BACKDROP                                          | gremove_backdrop                 |
| REMOVE SET OF NAMES FROM NAMESET                         | gremove_set_names_nameset        |
| REMOVE SET OF NAMES FROM NDC PICTURE                     | gremove_set_names_ndc_pic        |
| REMOVE SET OF SCISSORS FROM NDC PICTURE                  | gremove_set_scissors_ndc_pic     |
| REMOVE SET OF SCISSORS FROM SCISSOR SET                  | gremove_set_scissors_scissor_set |
| RENAME PICTURE PART                                      | grename_pic_part                 |
| RENAME SEGMENT                                           | grename_seg                      |
| RENAME STENCIL                                           | grename_stencil                  |
| RENAME TILING                                            | grename_tiling                   |
| REOPEN PICTURE PART                                      | greopen_pic_part                 |
| REORDER NDC PICTURE                                      | greorder_ndc_pic                 |
| REQUEST CHOICE                                           | greq_choice                      |
| REQUEST INPUT                                            | greq_in                          |
| REQUEST LOCATOR                                          | greq_loc                         |
| REQUEST PICK                                             | greq_pick                        |
| REQUEST STRING                                           | greq_string                      |
| REQUEST STROKE                                           | greq_stroke                      |
| REQUEST VALUATOR                                         | greq_val                         |
| RESET SPECIFIC WORKSTATION DESCRIPTION TABLE ENTRY STATE | greset_specif_ws_dt_entry_st     |
| RESTORE GKS STATE LIST                                   | grestore_gks_sl                  |
| RESTORE WORKSTATION STATE LIST                           | grestore_ws_sl                   |
| RETRIEVE PICTURE PART FROM ARCHIVE                       | gret_pic_part_ar                 |
| ROUTE                                                    | groute                           |
| SAMPLE CHOICE                                            | gsample_choice                   |
| SAMPLE INPUT                                             | gsample_in                       |
| SAMPLE LOCATOR                                           | gsample_loc                      |
| SAMPLE PICK                                              | gsample_pick                     |
| SAMPLE STRING                                            | gsample_string                   |
| SAMPLE STROKE                                            | gsample_stroke                   |
| SAMPLE VALUATOR                                          | gsample_val                      |
| SAVE GKS STATE LIST                                      | gsave_gks_sl                     |
| SAVE WORKSTATION STATE LIST                              | gsave_ws_sl                      |
| SELECT NORMALIZATION TRANSFORMATION                      | gsel_norm_tran                   |
| SET AREA INDEX                                           | gset_area_ind                    |
| SET AREA REPRESENTATION                                  | gset_area_rep                    |
| SET ATTRIBUTE SOURCE FLAGS                               | gset_asfs                        |
| SET CHARACTER CODE                                       | gset_char_code                   |
| 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 COLOUR REPRESENTATION                                | gset_colr_rep                    |
| SET CONTOUR ATTRIBUTE                                    | gset_cont_attr                   |
| SET DETECTABILITY                                        | gset_det                         |

|                                         |                                      |
|-----------------------------------------|--------------------------------------|
| SET EDGE COLOUR INDEX                   | <code>gset_edge_colr_ind</code>      |
| SET EDGE COLOUR SPECIFIER               | <code>gset_edge_colr_specif</code>   |
| SET EDGE FLAG                           | <code>gset_edge_flag</code>          |
| SET EDGETYPE                            | <code>gset_edgetype</code>           |
| SET EDGEWIDTH SCALE FACTOR              | <code>gset_edgewidth</code>          |
| SET ERROR HANDLING                      | <code>gset_err_hand</code>           |
| SET FONT INDEX MAPPING                  | <code>gset_font_ind_map</code>       |
| SET GLOBAL TRANSFORMATION MATRIX        | <code>gset_global_tran_matrix</code> |
| SET HIGHLIGHTING                        | <code>gset_highl</code>              |
| SET INTERIOR COLOUR INDEX               | <code>gset_int_colr_ind</code>       |
| SET INTERIOR COLOUR SPECIFIER           | <code>gset_int_colr_specif</code>    |
| SET INTERIOR STYLE                      | <code>gset_int_style</code>          |
| SET INTERIOR STYLE INDEX                | <code>gset_int_style_ind</code>      |
| SET LINE COLOUR INDEX                   | <code>gset_line_colr_ind</code>      |
| SET LINE COLOUR INDEX                   | <code>gset_marker_colr_ind</code>    |
| SET LINE COLOUR SPECIFIER               | <code>gset_line_colr_specif</code>   |
| SET LINE INDEX                          | <code>gset_line_ind</code>           |
| SET LINE REPRESENTATION                 | <code>gset_line_rep</code>           |
| SET LINETYPE                            | <code>gset_linetype</code>           |
| SET LINEWIDTH SCALE FACTOR              | <code>gset_linewidth</code>          |
| SET LOCAL TRANSFORMATION MATRIX         | <code>gset_local_tran_matrix</code>  |
| SET LOCATOR MODE                        | <code>gset_loc_mode</code>           |
| SET LOGICAL INPUT DEVICE MODE           | <code>gset_in_dev_mode</code>        |
| SET MARKER COLOUR SPECIFIER             | <code>gset_marker_colr_specif</code> |
| SET MARKER INDEX                        | <code>gset_marker_ind</code>         |
| SET MARKER REPRESENTATION               | <code>gset_marker_rep</code>         |
| SET MARKER SIZE SCALE FACTOR            | <code>gset_marker_size</code>        |
| SET MARKER TYPE                         | <code>gset_marker_type</code>        |
| SET NAMESET                             | <code>gset_nameset</code>            |
| SET NDC PICTURE PRIMITIVE ATTRIBUTE     | <code>gset_ndc_pic_prim_attr</code>  |
| SET NORMALIZATION TRANSFORMATION NUMBER | <code>gset_norm_tran_num</code>      |
| SET PATTERN REFERENCE POINT             | <code>gset_pat_ref_point</code>      |
| SET PATTERN REPRESENTATION              | <code>gset_pat_rep</code>            |
| SET PATTERN SIZE                        | <code>gset_pat_size</code>           |
| SET PICK IDENTIFIER                     | <code>gset_pick_id</code>            |
| SET PICK MODE                           | <code>gset_pick_mode</code>          |
| SET PRIMITIVE ATTRIBUTE                 | <code>gset_prim_attr</code>          |
| SET REPRESENTATION                      | <code>gset_rep</code>                |
| SET SCISSOR MODE                        | <code>gset_scissor_mode</code>       |
| SET SCISSOR SET                         | <code>gset_scissor_set</code>        |
| SET SEGMENT ATTRIBUTE                   | <code>gset_seg_attr</code>           |
| SET SEGMENT PRIORITY                    | <code>gset_seg_pri</code>            |
| SET SEGMENT TRANSFORMATION              | <code>gset_seg_tran</code>           |
| SET STENCIL ATTRIBUTE                   | <code>gset_stencil_attr</code>       |
| SET STRING MODE                         | <code>gset_string_mode</code>        |
| SET STROKE MODE                         | <code>gset_stroke_mode</code>        |
| SET TEXT ALIGNMENT                      | <code>gset_text_align</code>         |
| SET TEXT COLOUR INDEX                   | <code>gset_text_colr_ind</code>      |
| SET TEXT COLOUR SPECIFIER               | <code>gset_text_colr_specif</code>   |
| SET TEXT FONT AND PRECISION             | <code>gset_text_font_prec</code>     |
| SET TEXT INDEX                          | <code>gset_text_ind</code>           |
| SET TEXT PATH                           | <code>gset_text_path</code>          |
| SET TEXT REPRESENTATION                 | <code>gset_text_rep</code>           |
| SET TEXT SKEW ANGLE                     | <code>gset_text_skew_angle</code>    |
| SET VALUATOR MODE                       | <code>gset_val_mode</code>           |

|                                     |                                    |
|-------------------------------------|------------------------------------|
| SET VIEW                            | <code>gset_view</code>             |
| SET VIEW PRIORITY                   | <code>gset_view_pri</code>         |
| SET VIEW SELECTION CRITERION        | <code>gset_view_sel_crit</code>    |
| SET VIEWPORT                        | <code>gset_vp</code>               |
| SET VIEWPORT INPUT PRIORITY         | <code>gset_vp_in_pri</code>        |
| SET VISIBILITY                      | <code>gset_vis</code>              |
| SET WINDOW                          | <code>gset_win</code>              |
| SET WINDOW AND VIEWPORT             | <code>gset_win_vp</code>           |
| SET WORKSTATION SELECTION CRITERION | <code>gset_ws_sel_crit</code>      |
| SET WORKSTATION VIEWPORT            | <code>gset_ws_vp</code>            |
| SET WORKSTATION VISUAL EFFECTS      | <code>gset_ws_vis_effects</code>   |
| SET WORKSTATION WINDOW              | <code>gset_ws_win</code>           |
| SET WORKSTATION WINDOW AND VIEWPORT | <code>gset_ws_win_vp</code>        |
| TEXT                                | <code>gtext</code>                 |
| WRITE USER RECORD TO AUDIT          | <code>gwrite_user_rec_audit</code> |

5 Type definitions

5.1 Mapping of GKS data types

The GKS 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 8651.

| GKS data type               | C binding data type |
|-----------------------------|---------------------|
| Archive name                | Gint                |
| Archive specifier           | void *              |
| Audit identifier            | Gint                |
| Audit specifier             | void *              |
| Audit user data             | Gaudit_user_data    |
| Character code              | char                |
| Colour characteristics      | Gcolr_chars         |
| Error file                  | char *              |
| Escape function identifier  | Gint                |
| Escape input data record    | Gescape_in_data     |
| Escape output data record   | Gescape_out_data    |
| GDP data record             | Ggdp_data           |
| GDP function identifier     | Gint                |
| Glyph name                  | Gint                |
| ISO/IEC 9541 font name      | Gint                |
| Input data record           | Gin_data            |
| Integer                     | Gint                |
| Measure process identifier  | Gint                |
| NDC metafile specifier      | void *              |
| Name                        | Gint                |
| Nonnegative integer         | Gint                |
| Pick identifier             | Gint                |
| Picture identifier          | Gint                |
| Picture part name           | Gint                |
| Real number                 | Gfloat              |
| Realized metafile specifier | void *              |
| Scissor identifier          | Gint                |
| Segment name                | Gint                |
| Stencil name                | Gint                |
| Tiling name                 | Gint                |
| Trigger process identifier  | Gint                |
| Workstation generic type    | Gint                |
| Workstation identifier      | Gint                |
| Workstation specifier       | void *              |

5.2 Environment-defined type definitions

The data types defined in this clause allow for the ease of porting GKS/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/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/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/C. Suggest:

```
typedef int Gint;
```

**size\_t**

An additional environment-defined data type is the type **size\_t**. Depending on the environment, **size\_t** can be defined as **unsigned int** or **unsigned long**. **size\_t** is defined in the headers **<stdio.h>**, **<stdlib.h>**, **<stddef.h>**, **<string.h>**, **<time.h>**.

### 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, several implementation and registration dependent parts are of type **Gdata**. They can be replaced by any other structure.
- The data of registered prompt and echo types are represented by structures with the names **pet\_rn** (n the prompt and echo type), whereas the data of unregistered prompt and echo types are represented by structures with the names **pet\_un** (-n the prompt and echo type).
- The data of registered and unregistered GDPs, escape input and escape output are denoted in a similar way.
- For the input data, the mandatory part, as specified in GKS (ISO/IEC 7942-1), 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 defines prompt and echo type 2 this way for the CHOICE device;

---

**Gaudit\_user\_data** audit user data record

```
typedef struct {
    Gint      id;          /* identifier          */
    union {
        Gdata  impl_dep;   /* impl. dependent    */
    } data;
} Gaudit_user_data;
```

## Gchoice\_data choice data

```

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_u1 {
            Gdata      impl_dep;        /*      impl. dep. */
        } pet_u1; /* pet -1 data */
        /*. . . impl. defined PET's */
    } pets;
} Gchoice_data;

```

## Gcolr\_chars colour characteristics

```

typedef struct {
    enum Gcolr_chars_type { GTYPE_CIELUV, GTYPE_OTHER } type;
    union {
        Gcieluv  prim_colr[3]; /*      CIELUV primary colours */
        Gdata     impl_dep;    /*      impl. dep. */
    } chars;
} Gcolr_chars;

```

---

**Gescape\_in\_data** escape input data record

---

```
typedef union {
    struct Gescape_in_r1 {
        Gdata impl_dep; /* impl. 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 impl_dep; /* impl. 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 impl_dep; /* impl. dep. */
    } gdp_r1; /* gdp 1 data */
    struct Ggdp_u1 {
        Gdata impl_dep; /* impl. dep. */
    } gdp_u1; /* gdp -1 data */
    /* etc. */
} Ggdp_data;
```

---

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 attr_ctrl_flag;
            /* attribute control flag */
            Gline_attrs     line_attrs;
            /* line attributes */
            Gdata          impl_dep;      /* impl. dep. */
        } pet_r4;          /* pet 4 data */
        struct Gloc_pet_r5 {
            Gattr_ctrl_flag attr_ctrl_flag;
            /* attribute control flag */
            Gline_fill_ctrl_flag line_fill_ctrl_flag;
            /* line/area control flag */
            union Gloc_attrs {
                Gline_attrs line_attrs;
                /* line attrs. */
                Gareas_attrs area_attrs;
                /* interior 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_u1 {
            Gdata          impl_dep;      /* impl. dep. */
        } pet_u1;          /* pet -1 data */
        /* ... impl. defined PET's */
    } pets;
} Gloc_data;
```

---

Gpick\_data pick data

```

typedef struct {
    union Gpick_pets {
        struct Gpick_pet_r1 {
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r1;    /* pet 1 data */
        struct Gpick_pet_r2 {
            Gint        pick_id;      /* pick identifier */
            Gint        seg_name;     /* segment name */
            Gnameset    set_names;    /* set of names */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r2;    /* pet 2 data */
        struct Gpick_pet_r3 {
            Gnameset    set_names;    /* set of names */
            Gdata      impl_dep;      /* impl. dep. */
        } pet_r3;    /* pet 3 data */
        struct Gpick_pet_u1 {
            Gdata      data;          /* data */
        } pet_u1;    /* pet -1 data */
        /* etc. */
    } pets;
} Gpick_data;

```

---

## Gstring\_data string data record

```

typedef struct {
    Gint  in_buf_size;    /* input buffer size (nr. of bytes) */
    Gint  init_pos;      /* initial [cursor] position */
    union Gstring_pets {
        struct Gstring_pet_r1 {
            Gdata impl_dep;      /* impl. dep. */
        } pet_r1;    /* pet 1 data */
        struct Gstring_pet_u1 {
            Gdata impl_dep;      /* impl. dep. */
        } pet_u1;    /* pet -1 data */
        /* etc. */
    } pets;
} Gstring_data;

```

Gstroke\_data stroke data

```
typedef struct {
    Gint    init_pos;      /* initial buffer editing position */
    Gint    in_buf size;   /* input buffer size (nr. of points) */
    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;
            /* marker 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;
            /* line 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;
```

---

**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;

```

**5.4 Implementation independent type definitions**


---

**Gabut\_specif ABUT specifier**

```

typedef enum {
    GABUT_RIGHT,
    GABUT_LEFT,
    GABUT_TOP,
    GABUT_BOTTOM,
    GABUT_CTR_VERT,
    GABUT_CTR_HOR,
    GABUT_CAP_LINE,
    GABUT_BASE_LINE
} Gabut_specif;

```

**Garea\_attrs area attributes**


---

```

typedef struct {
    Gasf        int_style_asf;    /* interior style ASF */
    Gasf        style_ind_asf;    /* style index ASF */
    Gasf        colr_ind_asf;    /* colour specifier ASF */
    Gasf        edge_flag_asf;    /* edge flag ASF */
    Gasf        edgetype_asf;    /* edgetype ASF */
    Gasf        edgewidth_asf;    /* edgewidth scale factor ASF */
    Gasf        edge_colr_ind_asf; /* edge colour specifier ASF */
    Gint        ind;    /* area index */
    Garea_bundle bundle;    /* area bundle */
} Garea_attrs;

```

---

Garea\_bundle area bundle

```
typedef struct {
    Gint_style    int_style;           /* interior style          */
    Gint          style_ind;           /* interior style index    */
    Gcolr_kind    colr_kind;           /* interior colour kind    */
    Gint          colr_ind;            /* interior colour index   */
    Gcolr_rep     int_dir_colr;        /* direct interior colour  */
    Gedge_flag    edge_flag;           /* edge flag               */
    Gint          edgetype;            /* edgetype                */
    Gfloat         edgewidth;          /* edgewidth scale factor  */
    Gcolr_kind    edge_colr_kind;      /* edge colour kind        */
    Gint          edge_colr_ind;       /* edge colour index       */
    Gcolr_rep     edge_dir_colr;       /* direct edge colour      */
} Garea_bundle;
```

---

Garea\_bundle\_table area bundle table

```
typedef struct {
    Gint          num_inds;             /* number of area indices  */
    Gint          *inds;                /* list of area indices    */
    Garea_bundle  *bundles;             /* list of area bundles    */
} Garea_bundle_table;
```

---

Garea\_fac area facilities

```
typedef struct {
    Gint          num_int_styles;        /* number of interior styles */
    Gint_style    int_styles [5];        /* list of available interior styles */
    Gint_list     hatch_styles;          /* list of available hatch styles */
    Gint          num_pred_inds;         /* number of predef. area indices */
    Gint_list     edgetypes;             /* list of edgetypes        */
    Gint          num_edgewidths;        /* number of available edgewidths */
    Gfloat         nom_edgewidth;        /* nominal edgewidth        */
    Gfloat         min_edgewidth;        /* min. edgewidth           */
    Gfloat         max_edgewidth;        /* max. edgewidth           */
} Garea_fac;
```

---

Gasf attribute source flag

```
typedef enum {
    GASF_BUNDLED,
    GASF_INDIV
} Gasf;
```

---

**Gasfs attribute source flags**

```
typedef struct {
    Gasf    linetype;           /* linetype ASF                */
    Gasf    linewidth;         /* linewidth scale factor ASF   */
    Gasf    line_colr_ind;      /* line colour ASF              */
    Gasf    marker_type;        /* marker type ASF              */
    Gasf    marker_size;        /* marker size scale factor ASF */
    Gasf    marker_colr_ind;     /* marker colour ASF            */
    Gasf    text_font_prec;      /* text font and precision ASF  */
    Gasf    char_expan;         /* character expansion factor ASF */
    Gasf    char_space;         /* character spacing ASF        */
    Gasf    text_colr_ind;       /* text colour ASF              */
    Gasf    int_style;          /* interior style ASF           */
    Gasf    int_style_ind;       /* interior style index ASF      */
    Gasf    int_colr_ind;        /* interior colour ASF          */
    Gasf    edge_flag;          /* edge flag ASF                */
    Gasf    edgetype;           /* edgetype ASF                 */
    Gasf    edgewidth;          /* edgewidth scale factor ASF   */
    Gasf    edge_colr_ind;       /* edge colour ASF              */
} Gasfs;
```

---

**Gattr\_ctrl\_flag attribute control flag**

```
typedef enum {
    GFLAG_CUR,
    GFLAG_SPECIF
} Gattr_ctrl_flag;
```

---

**Gaudit\_flag audit flag**

```
typedef enum {
    GAUDIT_OFF,
    GAUDIT_ON
} Gaudit_flag;
```

---

**Gaudit\_op audit operation**

```
typedef struct {
    enum Gaudit_op_type {
        GAUDIT_OPEN, GAUDIT_CLOSE, GAUDIT_BEGIN, GAUDIT_END
    } type;
    void *specif; /* audit specifier */
} Gaudit_op;
```

---

Gbasic\_in\_data basic input data

```
typedef struct {
    Gin_class class; /* basic input [measure] class */
    union {
        Gloc_data      loc_data;      /* locator data */
        Gstroke_data   stroke_data;    /* stroke data */
        Gval_data       val_data;       /* valuator data */
        Gchoice_data    choice_data;    /* choice data */
        Gpick_data      pick_data;      /* pick data */
        Gstring_data    string_data;    /* string data */
    } in_data; /* input data */
} Gbasic_in_data;
```

---

Gbasic\_in\_value basic input value

```
typedef struct {
    Gin_class class; /* basic input [measure] class */
    union {
        Gloc      loc;      /* locator value */
        Gstroke    stroke;    /* stroke value */
        Gfloat     val;      /* valuator value */
        Gchoice    choice;    /* choice value */
        Gpick      pick;      /* pick value */
        char       *string;   /* string value */
    } in_value; /* input value */
} Gbasic_in_value;
```

---

Gbdry boundary

```
typedef struct {
    enum Gbdry_type {
        GBNDRY_CLOSED_PATH,
        GBNDRY_ELL_DISC
    } type;
    union {
        Gpath_list      closed_path_seq; /* closed path sequence */
        Gconic_matrix    ell_disc;       /* elliptic disc */
    } value;
} Gbdry;
```

---

Gbdry\_list boundary list

```
typedef struct {
    Gint      num_bndries; /* number of boundaries in list */
    Gbdry     *bndries;    /* list of boundaries */
} Gbdry_list;
```

---

**Gcap cap**

```
typedef enum {
    GCAP_BUT,
    GCAP_ROUND,
    GCAP_SQUARE
} Gcap;
```

---

**Gchange\_flag change flag**

```
typedef enum {
    GFLAG_CHANGED,
    GFLAG_NOT_CHANGED
} Gchange_flag;
```

---

**Gchoice choice [value]**

```
typedef struct {
    Gchoice_status    status;           /* status          */
    Gint              num;              /* choice number   */
} Gchoice;
```

---

**Gchoice\_status choice status**

```
typedef enum {
    GCHOICE_OK,
    GCHOICE_NO_CHOICE
} Gchoice_status;
```

---

**Gcieluv CIE L\*u\*v\* [colour specification]**

```
typedef struct {
    Gfloat    u_prime;    /* u' coefficient */
    Gfloat    v_prime;    /* v' coefficient */
    Gfloat    y;          /* y luminance    */
} Gcieluv;
```

---

**Gclip\_ind clipping indicator**

```
typedef enum {
    GIND_NO_CLIP,
    GIND_CLIP
} Gclip_ind;
```

---

**Gclosed\_path\_op\_st closed path operating state**

```
typedef enum {
    GCP_ST_CPCL,
    GCP_ST_CPOP
} Gclosed_path_op_st;
```

---

Gcolr\_avail colour available

---

```
typedef enum {
    GAVAIL_MONOCHR,
    GAVAIL_COLR
} Gcolr_avail;
```

---

Gcolr\_fac colour facilities

---

```
typedef struct {
    Gint      num_colr_models; /* number of colour models */
    Gint      colr_models[4]; /* list of available colour models */
    Gint      num_colrs; /* number of colours */
    Gcolr_chars chars; /* colour characteristics */
    Gcolr_avail colr_avail; /* colour availability */
    Gint      num_pred_inds; /* number of predef. colour indices */
} Gcolr_fac;
```

---

Gcolr\_kind colour kind

---

```
typedef enum {
    GCOLR_INDIR,
    GCOLR_RGB,
    GCOLR_CIELUV,
    GCOLR_HLS,
    GCOLR_HSV
} Gcolr_kind;
```

---

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 */
              /* etc. */
} Gcolr_rep;
```

---

Gcolr\_specif colour specifier

---

```
typedef struct {
    Gcolr_kind kind; /* colour kind */
    union {
        Gint      ind; /* colour index */
        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 */
    } value;
} Gcolr_specif;
```

---

**Gcolr\_table** colour table

---

```
typedef struct {
    Gint      num_inds;          /* number of colour indices */
    Gint      *inds;            /* list of colour indices */
    Gcolr_specif *colr_specifs; /* list of colour specifications */
} Gcolr_table;
```

---

**Gcomp\_data** composite data

---

```
typedef struct {
    Gint      num_devs;          /* number of [basic] devices */
    Gbasic_in_data *basic_in_data; /* list of basic input data */
} Gcomp_data;
```

---

**Gcomp\_value** composite value

---

```
typedef struct {
    Gint      num_devs;          /* number of [basic] devices */
    Gbasic_in_value *basic_in_value; /* list of basic input values */
} Gcomp_value;
```

---

**Gconic\_matrix** conic matrix

---

```
typedef Gfloat Gconic_matrix[3] [3];
```

---

**Gconic\_sec** conic section

---

```
typedef struct {
    Gconic_matrix      matrix;          /* conic matrix */
    Gpoint              point_1;         /* point # 1 */
    Gpoint              point_2;         /* point # 2 */
    Gsense_flag         sense_flag;      /* sense flag */
} Gconic_sec;
```

---

**Gconic\_sec\_list** conic primitive list

---

```
typedef struct {
    Gint      num_conic_prims; /* number of conic primitives in list */
    Gconic_sec *conic_prims;   /* list of conic primitives */
} Gconic_sec_list;
```

---

**Gcont\_attr\_name** contour attribute name

---

```
typedef enum {
    GNAME_STYLE,
    GNAME_WIDTH,
    GNAME_CAP,
    GNAME_JOIN
} Gcont_attr_name;
```

---

---

Gcont\_attr\_value contour attribute value

```
typedef struct {
    Gcont_attr_name    name;        /* attribute name    */
    union {
        Gint           style;       /* style            */
        Gfloat          width;      /* width            */
        Gcap            cap;        /* cap              */
        Gjoin           join;       /* join             */
    } value;
} Gcont_attr_value;
```

---

Gcont\_attrs contour attributes

```
typedef struct {
    Gint    style;    /* style    */
    Gfloat  width;    /* width    */
    Gcap    cap;      /* cap      */
    Gjoin   join;     /* join     */
} Gcont_attrs;
```

---

Gconv\_flag conversion flag

```
typedef enum {
    GCONV_DONE,
    GCONV_APPROX,
    GCONV_NOT_CONV
} Gconv_flag;
```

---

Gcoord\_switch coordinate switch

```
typedef enum {
    GCOORD_WC,
    GCOORD_NDC
} Gcoord_switch;
```

---

Gctrl\_flag control flag

```
typedef enum {
    GFLAG_COND,
    GFLAG_ALWAYS
} Gctrl_flag;
```