

SYMBOLIC

COMPUTATION

G. Enderle K. Kansy G. Pfaff

# Computer Graphics Programming

GKS – The Graphics Standard

Second, Revised and Enlarged Edition



Springer-Verlag

G. Enderle K. Kansy G. Pfaff

# Computer Graphics Programming

GKS - The Graphics Standard

Second, Revised and Enlarged Edition

With 100 Figures, Some in Color



Springer-Verlag  
Berlin Heidelberg New York  
London Paris Tokyo

**Dr. Günter Enderle** †

Standard Elektrik Lorenz AG  
Lorenzstraße 10, 7000 Stuttgart 40, FRG

**Dr. Klaus Kansy**

Gesellschaft für Mathematik und Datenverarbeitung  
Schloss Birlinghoven, 5205 St. Augustin 1, FRG

**Dr. Günther Pfaff**

GTS-GRAL Graphische Standards für Computer-Systeme GmbH  
Alsfelder Straße 7, 6100 Darmstadt, FRG

ISBN 3-540-16317-4 Springer-Verlag Berlin Heidelberg New York  
ISBN 0-387-16317-4 Springer-Verlag New York Berlin Heidelberg

ISBN 3-540-11525-0 1. Auflage Springer-Verlag Berlin Heidelberg New York Tokyo  
ISBN 0-387-11525-0 1st edition Springer-Verlag New York Heidelberg Berlin Tokyo

Library of Congress Cataloging-in-Publication Data.

Enderle, G. (Günter), Computer graphics programming. (Symbolic computation. Computer graphics)

Bibliography: p. Includes index. I. Computer graphics—Standards. I. Kansy, K. (Klaus) II. Pfaff, G. (Günther) III. Title. IV. Series. T385.E53 1986 006.6'0218 86-20370

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this publication or parts thereof is only permitted under the provisions of the German Copyright Law of September 9, 1965, in its version of June 24, 1985, and a copyright fee must always be paid. Violations fall under the prosecution act of the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1984, 1987  
Printed in Germany

Typesetting, printing, and bookbinding: Universitätsdruckerei H. Stürtz AG, Würzburg  
2145/3140-543210

## **In Memoriam Günter Enderle**

Shortly before the second edition of this book went into print, we received the message that our friend and co-author of this book, Günter Enderle, died in a car accident on January 13, 1987, at the age of 42.

Günter Enderle received his Dipl.-Ing. (M.S.) in 1971 and Dr.-Ing. (Ph.D. in Engineering) in 1975 from the University of Karlsruhe. He was with the Karlsruhe Nuclear Research Centre from 1971 and became group leader for Computer Graphics and CAD. From 1984 onwards, he was responsible for software development at Standard Elektrik Lorenz (SEL) in Stuttgart. He was a member of EUROGRAPHICS from 1980. He held the position of editor-in-chief of the journal Computer Graphics Forum and inaugurated the book series Eurographic Seminars — Tutorials and Perspectives in Computer Graphics.

The Graphical Kernel System GKS was a focal point of his professional interests. He contributed significantly to its design as a member, from 1979, of the editorial board and by creative participation in the international review process. From 1981 he was chairman of the German Standardization Committee DIN-NI-5.9 (later renamed DIN-NI-21.2) “Computer Graphics” and head of the German delegation to the ISO Working group ISO/TC 97/SC 5/WG 2 (later renamed ISO/TC 97/SC 21/WG 2). Günter Enderle performed these tasks with great energy. He had the gift of motivating people participating in the different committees as volunteers to put all their force into the promotion of a common goal. Therefore, the success of the Graphical Kernel System GKS is closely related to his name.

For nearly ten years, the undersigned collaborated closely with Günter Enderle. For the second author (K.K.), this cooperation started 1978 within the Coordinating Committee “Computer Graphics” of the West German Association of National Research Centres (AGF) with the definition of the so-called AGF-Plotfile, a predecessor of the GKS Metafile.

All three authors came together through their membership of the GKS editorial board and had a very fruitful and productive time with the development of numerous versions of the GKS proposal and in dealing with the bulk of comments which came

in when GKS was presented to the international standardization bodies and which had to be handled before GKS was accepted as an ISO work item. In this task, Günter Enderle proved his ability in finding solutions for difficult problems and in realizing solutions within a short time.

The work around GKS included numerous meetings, national and international. These meetings were primarily devoted to hard technical work. Besides and through this technical work, “a network of deep friendship and common understanding has been established”, as Günter Enderle himself expressed it on page 59 of this book. Therefore, we have not only lost a creative and dedicated colleague, but also a close friend, with whom we shared work and leisure time for many years. We are sure that our sorrow will be shared by all the colleagues who met Günter Enderle in the various German and international standardization committees for Computer Graphics.

Bonn, Darmstadt, February 1987

*Klaus Kansy  
Günther Pfaff*

## FOREWORD TO THE SECOND EDITION

When this book was published in 1983, the process of designing the Graphical Kernel System (GKS) was in its final stages. The final version of the first international standard for Computer Graphics was expected before the end of 1983. However, finalizing a standard is a complex and time-consuming process, so that the International Standard version of GKS appeared in August 1985. Before, the final letter ballot on GKS had been conducted. Comments raised in the ballot by the National Standardization Bodies have led to a number of small changes in the document. The final version of the GKS document was prepared in 1984 and forwarded to ISO central office for publication as an ISO standard. A number of GKS language bindings have also reached a stable and reliable status, and their ISO standard versions are expected soon.

Once GKS had been accepted and recognized as the cornerstone of a family of compatible graphics standards, a number of new projects were started with the aim of standardizing additional important graphics interfaces. The changes in GKS, the finalization of the GKS language bindings, the new projects in the computer graphics field, and last but not least, the extraordinary success of the first edition of this book, have led us to prepare this second edition.

The main differences from the first edition are as follows:

- All changes incorporated in GKS as a result of the final ISO letter ballot have been included.
- The language bindings, primarily FORTRAN and Pascal, have been updated to reflect the latest versions of the GKS Language Bindings Standard.
- The Pascal examples have been modified to reflect the traditional printed appearance of Pascal programs (e.g., keywords of the language are printed in bold letters).
- The three-dimensional (3D) extension of GKS has been completely revised. A new part (Part IV) presents in detail the proposed GKS-3D Standard which adds a complete set of 3D functions to GKS.

- The new projects on a Computer Graphics Metafile (CGM) and on Computer Graphics Interfaces for graphical devices (CGI) have been taken into consideration, where appropriate.

We are confident that this second edition will reinforce the original goal of this book, namely to offer a complete reference for understanding, using, learning, teaching, and implementing the Graphical Kernel System and its environment.

Karlsruhe, Bonn, Darmstadt, November 1986

*Günter Enderle*

*Klaus Kansy*

*Günther Pfaff*

## FOREWORD TO THE FIRST EDITION

For several years the authors of this book have been involved in the design and the national and international review of the forthcoming graphical standard. When it became apparent that this process was coming to an end and the International Standard "Graphical Kernel System" (GKS) was cast into its final form, an urgent need arose to provide the graphics community with detailed information on the new standard, and to educate graphics programmers in GKS. One major goal of GKS, besides the portability of graphical application programs and device independence, is "programmer portability" which it aims to achieve by establishing a common basis for the training of graphics programmers. Having taken part in the development of GKS from the very early stages of defining the basic concepts and designing its first versions up to the final draft of the International Standard, we felt it would be worthwhile embarking on the venture of writing a text book on computer graphics programming based on GKS.

This book is aimed, on the one hand, at graphics users, experts and managers who want to gain an overview of the new standard and a better understanding of its concepts. On the other hand, it addresses graphics programmers who want to use GKS for realizing their graphical applications. It can serve as a basis for teaching and studying the functions, concepts and methods of GKS. Additionally, it will be a valuable source of information for implementors of GKS.

One of the main areas of application of computer graphics is Computer Aided Design (CAD). GKS can serve as an excellent base upon which portable CAD systems can be built. A thorough introduction to CAD is presented in another book in the SYMBOLIC COMPUTATION series: "CAD — Fundamentals and System Architectures", by J. Encarnação and E.G. Schlechtendahl [ENCA 82a].

The standard document defining GKS has to be complete and consistent, it uses formal descriptions where possible, and it has to adhere to certain formal rules for the specification of the standard. Only rarely does it give informal introductions, examples, or explanations for the decisions taken. For an over-



view such as this book aims to provide, however, an informal and less complete presentation is more suitable and examples, figures and explanations are essential for teaching purposes. We want to offer easy access to GKS and to the graphics environment in which it is situated.

It is of course inevitable in a book describing a standard that some material from the standard document will be used. The authors were members of the editorial team which designed GKS up to version 4.8 [DIN 79], and since then have taken part in the development of further versions of the GKS document up to the GKS standard. However, the GKS document is the result of combined contributions from many different people. We feel that the quality of the document is very high, and that there is no better way of describing some aspects of the GKS standard. Accordingly, in such cases the relevant parts of the document are reproduced in this book with little or no alteration.

The book is divided into five main parts. In Part I, an overview of the integration of GKS into the Computer Graphics framework is given and the principles and basic concepts of GKS are introduced and explained. Part II describes the design process of GKS in the committee NI/UA-5.9 of the Deutsches Institut für Normung (DIN), and the extensive international review and refinement carried out by the experts of working group TC97/SC5/WG2 of the International Organization for Standardization (ISO). The groups participating in this process, important events, and major design decisions, are presented, as well as the methods used for handling the review and the revision of the standard draft and for resolving conflicts.

Part III of the book is devoted to explaining the GKS functions and their applications. All GKS functions and their parameters are described, both in the language-independent form presented in the standard document and in the FORTRAN subroutine version. Various examples are given in the programming languages FORTRAN and Pascal. Exercises are provided in order to deepen the reader's understanding of the functions and to assist in the teaching of Computer Graphics programming on the basis of GKS. In Part IV, the 3D extensions to GKS are described.

The last part of the book covers the various interfaces of the standard within the Computer Graphics environment. Before a standardized graphics system can be used, it has to be implemented on existing hardware and operating systems. It is of course desirable that such implementations be validated and that their conformity to the standard be certified. The implementation of the standard in a given programming language is made possible by adapting the standard functions to the rules of that language. We describe the FORTRAN binding of GKS. Further interfaces

exist with graphical input and output devices, and with graphics metafiles for storage of pictures. Communication between members of the graphics community is facilitated by a common graphics terminology developed in parallel with the GKS standard.

Newcomers to the Computer Graphics field and readers who want an overview of GKS concepts should read Part I and then use the table of contents or the index at the end of the book to find information of special interest to them. Part II addresses itself to those interested in standards and how they are created. Programmers and scientists designing graphics applications will find a detailed description of the GKS functions in Part III. This part will also be the main reference for learning Computer Graphics programming on the basis of GKS. Part V will be of special importance to GKS implementors.

We hope this book will help to disseminate the application of the Graphical Kernel System, to explain its principles and concepts, and to promote Computer Graphics education on the basis of the first standard in Computer Graphics.

Karlsruhe, Bonn, Darmstadt, July 1983

*Günter Enderle*

*Klaus Kansy*

*Günther Pfaff*

## DEDICATION

The subject of this book — the Graphical Kernel System — was developed in a long process starting in 1976 and has finally evolved as an International Standard. This book is dedicated to the graphics experts who designed GKS under the auspices of the Deutsches Institut für Normung (DIN), and who played a part in its evolution through the international review process within the working group TC97/SC5/WG2 “Computer Graphics” of the International Organization for Standardization (ISO) and various national standardization organizations. Over 100 scientists from all over the world invested more than 50 man-years in this venture, making GKS the consistent and complete graphics standard it is today, developing a concise terminology for Computer Graphics, establishing the firm basis of a methodology for Computer Graphics, and cooperating in a spirit of mutual confidence and friendship.

### Members and experts of DIN-NI/UA-5.9 “Computer Graphics” 1975—1982:

R. Anderl, E. Bauböck, H. Borik, H.-G. Borutka, L. Brandenburger, H. Brüggemann, P. Dobrowolski, R. Eckert, P. Egloff, J. Encarnação, G. Enderle, B. Fink, H. Flegel, R. Gnatz, M. Gonauser, H. Grauer, I. Grieger, Th. Johannsen, E. Jungmann, K. Kansy, R. Karg, W. Klingenberg, R. Konkart, H. Kuhlmann, G. Lang-Lendorff, St. Lewandowski, G. Mittelstraß, G. Nees, H. Nowacki, D. Otto, K. Pasemann, G. Pfaff, F.-J. Prester, K. Reumann, J. Rix, H.-J. Rosenberg, F.-K. Roth, E.G. Schlechtendahl, J. Schönhut, R. Schuster, D. Stroh, D. Völkel, J. Weiss, J. Weskott, H. Wetzel, P. Wißkirchen.

### Members and experts of ISO TC97/SC5/WG2 “Computer Graphics” 1977—1982:

J. Bettels (CH), K. Bö (N), P. Bono (USA), H.-G. Borutka (D), K. Brodie (UK), L. Brown (NL-C), F. Canfield (USA), St. Carson (USA), J. Chin (USA), U. Cugini (I), J. Daabeck (DK), G. Dettori (I), F. de Witte (NL), D. Duce (UK), A. Ducrot (F), R. Dunn (USA), R. Eckert (D), P. Egloff (D), J. Encarnação (D), G. Enderle (D), J. Ero (NL), D. Fisher (UK), A. Francis (UK), J. Gallop (UK), P. Gauriat (F), R. Gnatz (D), T.H. Gossling (UK), I. Grieger (D), R. Guedj (F), Y. Gueniot

XIV Dedication

(F), Ch. Hatfield (USA), W. Herzner (A), B. Herzog (USA), F.R.A. Hopgood (UK), M. Hosaka (J), K. Kansy (D), R. Kessener (NL), F. Kimura (J), J. Kivi (SF), A. Kotzauer (GDR), G. Krammer (H), H. Kuhlmann (D), R. Langridge (UK), M. Lucas (F), V. Lvov (USSR), J. Matthijs (B), J. Michener (USA), L. Moltedo (I), G. Nees (D), H. Newman (C), Ch. Osland (UK), Ch. Pellegrini (CH), G. Pfaff (D), A. Planman (SF), M. Polisher (USA), F.-J. Prester (D), R. Puk (USA), T. Reed (USA), K. Reumann (D), J. Rix (D), D.S.H. Rosenthal (UK), J. Rowe (USA), J. Schönhut (D), B. Shepherd (USA), E. Sonderegger (USA), R. Spiers (UK), D. Stroh (D), R. Sulonen (SF), D. Sutcliffe (UK), Z. Tolnay-Knefely (H), P. ten Hagen (NL), J. van der Star (NL), H. van Velden (NL), A. Warman (UK), M. Wein (C), J. Weiss (A), M. Whyles (USA), K. Willet (USA), A. Williams (UK), G. Williams (C), R. Williams (USA), P. Wißkirchen (D).

# CONTENTS

<b>Part I — Introduction to Computer Graphics Based on GKS</b>	<b>1</b>
1	<i>What Is Computer Graphics?</i> . . . . . 2
1.1	Definition of Computer Graphics . . . . . 2
1.2	Areas of Computer Graphics . . . . . 2
1.3	Impact of the Graphical Kernel System on Computer Graphics . . . . . 4
2	<i>Aims and Contents of Part I</i> . . . . . 4
3	<i>The Computer Graphics User</i> . . . . . 5
4	<i>Interfaces of the Graphical Kernel System</i> . . . . . 6
5	<i>Principles and Goals of the Graphical Kernel System</i> . . . . . 8
6	<i>Main Concepts of the Graphical Kernel System</i> . . . . . 11
7	<i>Creating Graphical Output</i> . . . . . 13
7.1	Line Graphics and Raster Graphics . . . . . 13
7.2	Output Primitives and Attributes . . . . . 14
7.2.1	Output Primitives . . . . . 15
7.2.2	Output Primitive Attributes . . . . . 16
7.3	Indices, Bundles, and Tables . . . . . 20
8	<i>Coordinate Systems and Transformations</i> . . . . . 25
8.1	Coordinate Systems . . . . . 25
8.2	Transformations . . . . . 26
8.3	The Normalization Transformation . . . . . 26
8.4	Clipping . . . . . 28
9	<i>The Graphical Workstation</i> . . . . . 28
9.1	Routing Output to Workstations . . . . . 29
9.2	Types of GKS Workstations . . . . . 30
9.3	The Workstation Transformation . . . . . 30
9.4	The Deferral State . . . . . 32
9.5	Addressing Special Workstation Capabilities . . . . . 32
10	<i>Input</i> . . . . . 33
10.1	Interactive Computer Graphics . . . . . 33
10.2	Logical Input Device Classes . . . . . 33
10.3	Operating Modes . . . . . 34
10.4	Echoes, Prompts, and Input Device Initialisation . . . . . 36

11	<i>Segments</i>	37
11.1	Structuring Pictures	37
11.2	Manipulating Segments	38
11.3	Segment Attributes	39
11.4	Segment Transformations	40
11.5	Workstation-Dependent and Workstation-Independent Segment Storage	40
11.6	Copying Segments	41
12	<i>The GKS Metafile</i>	42
12.1	Graphics Metafiles	42
12.2	GKSM and CGM	43
12.3	The Metafile Interface	45
12.4	The Metafile Formats	45
13	<i>States and State Lists</i>	46
13.1	The GKS Operating State	46
13.2	The State Lists	47
13.3	Inquiry Functions for State List Values	47
14	<i>Error Handling</i>	48
15	<i>3D Extensions to GKS</i>	48

**Part II — The Process of Generating a Standard** . . . . . 51

1	<i>The Evolution of Computer Graphics</i>	52
1.1	Graphical Devices	52
1.2	Graphics Software	53
1.3	SEILLAC I	55
2	<i>Committees, People, and Events</i>	57
2.1	Graphics Standardization Committees	57
2.2	People	59
2.3	Overview of the Main Events	59
3	<i>GKS Review — Problems and Their Solutions</i>	62
3.1	Scope of the Standard	65
3.2	Output Primitives	66
3.3	Segmentation	70
3.4	Input	74
3.5	Transformations	77
3.6	Workstation Concept	80

**Part III — Graphical Kernel System Programming** . . . . . 83

1	<i>Format and Structure of Part III</i>	84
1.1	Contents of Part III	84
1.2	Format of GKS Function Definitions	84
1.2.1	Example of the Definition of GKS Functions	85

1.2.2	Format of Language-Independent Definition . . . . .	85
1.2.3	Format of FORTRAN Definition . . . . .	87
1.3	Format of Examples . . . . .	88
1.3.1	Example of an Example . . . . .	88
1.3.2	Format of Pascal Examples . . . . .	88
1.3.3	Format of FORTRAN Examples . . . . .	89
1.4	Exercises . . . . .	89
2	<i>Levels</i> . . . . .	89
2.1	Overview . . . . .	89
2.2	Functionality of the GKS Levels . . . . .	92
2.2.1	The Minimal or Lowest Level of GKS (Level 0a) . . . . .	93
2.2.2	Level 0b . . . . .	96
2.2.3	Level 0c . . . . .	97
2.2.4	Levels 1 a, 1 b, 1 c . . . . .	99
2.2.5	Levels 2 a, 2 b, 2 c . . . . .	100
2.3	The Impact on Writing Portable Application Programs . . . . .	101
2.4	Exercises . . . . .	102
3	<i>States and State Lists</i> . . . . .	103
3.1	Introduction . . . . .	103
3.2	Operating States . . . . .	104
3.3	The Functions Allowed in Individual States . . . . .	107
3.4	State Lists . . . . .	109
3.5	Basic Control Functions . . . . .	115
3.6	Examples . . . . .	117
4	<i>Workstations</i> . . . . .	119
4.1	Introduction . . . . .	119
4.2	Workstation Description Table . . . . .	121
4.3	Workstation State List . . . . .	128
4.4	GKS Functions Which Involve Workstations . . . . .	132
4.5	Workstation Control . . . . .	138
4.6	Deferring Picture Changes . . . . .	143
4.7	Addressing Workstation Capabilities Not Covered by GKS . . . . .	148
4.8	Examples . . . . .	150
4.9	Exercises . . . . .	152
5	<i>Transformations</i> . . . . .	153
5.1	Coordinate Systems . . . . .	153
5.2	Normalization Transformation . . . . .	154
5.3	Workstation Transformation . . . . .	156
5.4	Clipping . . . . .	159
5.5	Transformation of LOCATOR and STROKE Input . . . . .	161
5.6	Examples . . . . .	163
5.7	Exercises . . . . .	171
6	<i>Output Primitives</i> . . . . .	172
6.1	Introduction . . . . .	172
6.2	POLYLINE Primitive . . . . .	180
6.2.1	Global POLYLINE Attributes . . . . .	181

6.2.2	Workstation-Dependent POLYLINE Attributes . . . . .	184
6.3	POLYMARKER Primitive . . . . .	186
6.3.1	Global Polymarker Attributes . . . . .	188
6.3.2	Workstation-Dependent POLYMARKER Attributes . . . . .	191
6.4	TEXT Primitive . . . . .	192
6.4.1	TEXT Attributes for Beginners . . . . .	193
6.4.2	Character Body . . . . .	195
6.4.3	Text Extent Rectangle . . . . .	196
6.4.4	Text Font and Precision . . . . .	199
6.4.5	Setting Global Text Attributes . . . . .	201
6.4.6	Workstation-Dependent TEXT Attributes . . . . .	206
6.5	FILL AREA Primitive . . . . .	208
6.5.1	Global FILL AREA Attributes . . . . .	210
6.5.2	Workstation-Dependent FILL AREA Attributes . . . . .	215
6.6	CELL ARRAY Primitive . . . . .	218
6.7	GENERALIZED DRAWING PRIMITIVE (GDP) . . . . .	220
6.8	Attribute Setting Functions Which Concern All Primitives . . . . .	222
6.8.1	Setting the Colour Table . . . . .	222
6.8.2	Setting the Aspect Source Flags . . . . .	224
6.8.3	Setting the Pick Identifier . . . . .	225
6.8.4	Clipping Rectangle . . . . .	226
6.9	Examples . . . . .	226
6.10	Exercises . . . . .	235
7	<i>Segments</i> . . . . .	237
7.1	Introduction . . . . .	237
7.2	How Are Segments Generated? . . . . .	237
7.3	What Is Stored in a Segment? . . . . .	238
7.4	When Are Primitives Taken from the Transformation Pipeline? . . . . .	239
7.5	Segment Creation and Deletion . . . . .	241
7.6	Manipulation of the Segment Attributes . . . . .	246
7.7	The Workstation-Independent Segment Storage (WISS) . . . . .	251
7.8	Different Levels of Segmentation . . . . .	257
7.9	Utility Functions . . . . .	257
7.10	Examples . . . . .	259
7.11	Exercises . . . . .	272
8	<i>Input</i> . . . . .	273
8.1	Introduction to Logical Input Devices . . . . .	273
8.1.1	Identification . . . . .	274
8.1.2	Logical Input Classes and Values . . . . .	274
8.1.3	Operating Modes . . . . .	275
8.1.4	Logical Input Device Model . . . . .	275
8.1.5	Setting the Logical Input Device Mode . . . . .	277
8.2	Details About Logical Input Devices . . . . .	278
8.2.1	LOCATOR and STROKE Devices . . . . .	278
8.2.2	VALUATOR . . . . .	281
8.2.3	CHOICE . . . . .	281



8.2.4	PICK	281
8.2.5	STRING	282
8.3	Initialising Logical Input Devices	282
8.4	Changing the Input Device Mode	294
8.5	Request Input	298
8.6	Sample Input Functions	310
8.7	Event Input	319
8.7.1	Input Queue and Current Event Report	319
8.7.2	Functions to Await and Delete Event Queue Entries	320
8.7.3	Get Input Functions	325
8.8	A Compound Example of Using the GKS Input Functions	330
8.9	Exercises	338
9	<i>Error Handling</i>	340
9.1	Strategy	340
9.2	The Emergency Closure Procedure	342
9.3	The ERROR HANDLING and ERROR LOGGING Procedures	342
9.4	Error Detection Within GKS Procedures	345
9.5	Reaction of Application Programs to Error Detections	345
9.6	List of GKS Error Numbers and Messages	346
9.6.1	States	346
9.6.2	Workstations	347
9.6.3	Transformations	348
9.6.4	Output Attributes	348
9.6.5	Output Primitives	349
9.6.6	Segments	349
9.6.7	Input	350
9.6.8	Metafiles	350
9.6.9	Escape	350
9.6.10	Miscellaneous Errors	350
9.6.11	System Errors	351
9.6.12	Reserved Errors	351
9.6.13	Implementation-Dependent Errors	351
9.7	Exercises	352
10	<i>Inquiry Functions</i>	352
10.1	State Lists and Inquiry Functions	352
10.1.1	Error Reports in Inquiry Functions	353
10.1.2	Inquiry Functions for the Settable State Lists	353
10.1.3	Inquiry Functions for the Workstation Description Table	354
10.1.4	Inquiry Functions for the Error State List and Pixel Memories	354
10.2	Description of the Inquiry Functions	354
10.2.1	Inquiry Function for Operating State	355
10.2.2	Inquiry Functions for GKS Description Table	355
10.2.3	Inquiry Functions for GKS State List	357
10.2.4	Inquiry Functions for Workstation State List	364
10.2.5	Inquiry Functions for Workstation Description Table	377