GH24-5008-1
File No. S370/4300-50

Data Language/I Disk Operating System/ Virtual Storage (DL/I DOS/VS) Library Guide and Master Index

Program Product

Program Number 5746-XX1

GH24-5008-1
File No. S370/4300-50

Data Language/I Disk Operating System/ Virtual Storage (DL/I DOS/VS) Library Guide and Master Index

Program Product

Program Number 5746-XX1

Second Edition (January 1984)

This edition, GH24-5008-1, applies to Version 1, Release 7 (Version 1.7) of Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS), Program Number 5746-XX1, and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography,* GC20-0001.

Summary of Changes

For a list of changes, see page iii.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

IBM Corporation
Dept. 812BP
1133 Westchester Avenue
White Plains, NY 10604 USA

or

IBM Deutschland GmbH
Dept. 3282
Schoenaicher Strasse 220
D-7030 Boeblingen
Federal Republic of Germany

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

Summary of Changes
for SH24-5008-1
Version 1.7

This edition has been revised to include two new books: *DL/I DOS/VS Interactive Resource Definition and Utilities,* SH24-5029, and *DL/I DOS/VS Recovery/Restart Guide,* SH24-5030. Also included is Interactive Utility Generation (IUG) online information.

This manual also includes an updated master index for the DL/I library as well as miscellaneous corrections and updates to existing information.

Preface

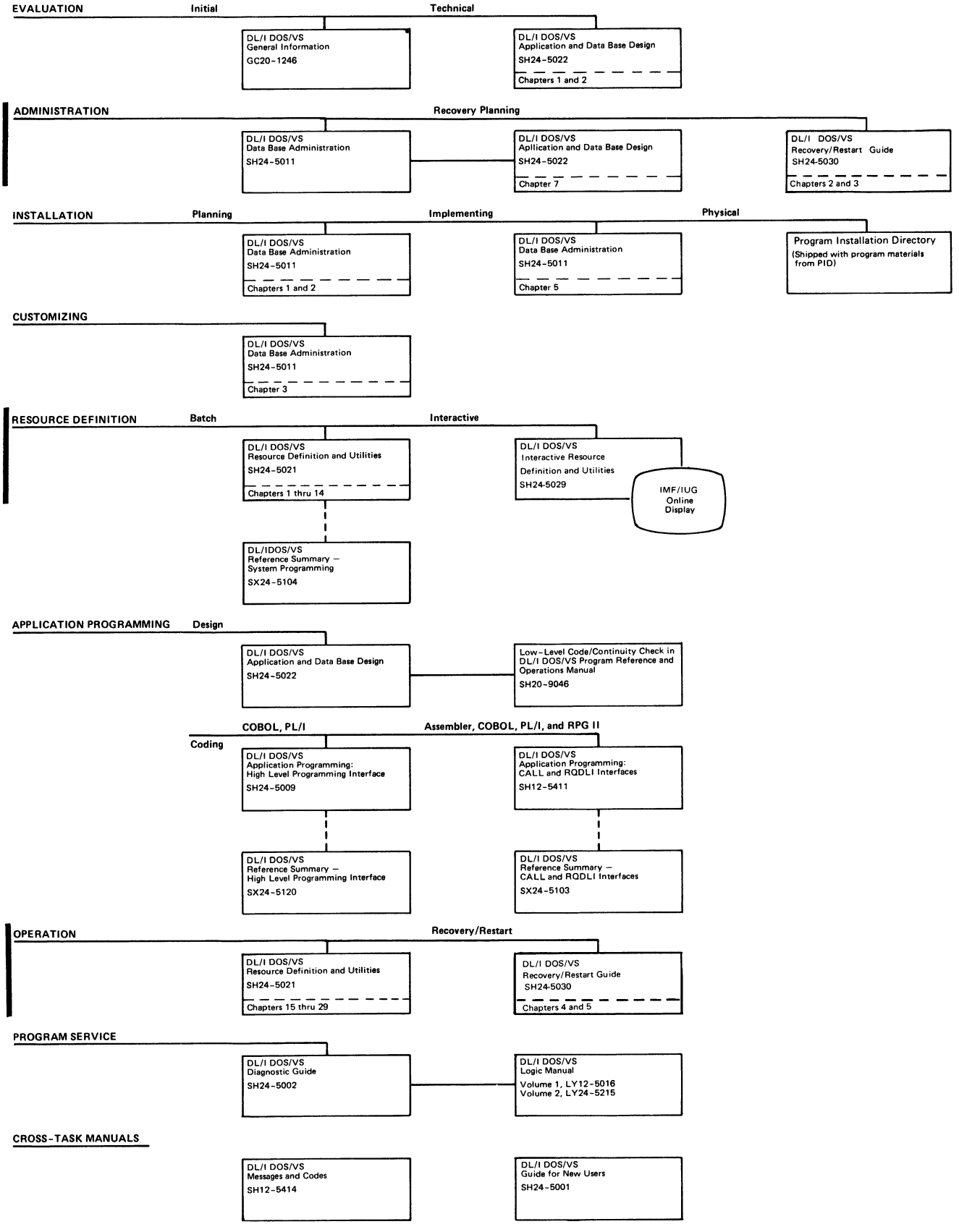


Figure 1. DL/I DOS/VS Library Organization

DL/I DOS/VS Publications

In order number sequence:

Order No.	Publication Title
GH20-1246	*DL/I DOS/VS General Information*
GH24-5008	*DL/I DOS/VS Library Guide and Master Index*
GH24-5031	*DL/I DOS/VS Program Product Specifications*
LY12-5016	*DL/I DOS/VS Logic, Volume 1*
LY24-5215	*DL/I DOS/VS Logic, Volume 2*
SH12-5411	*DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces*
SH12-5414	*DL/I DOS/VS Messages and Codes*
SH20-9046	*Low-Level Code/Continuity Check in Data Language/I DOS/VS Program Reference and Operations Manual*
SH24-5001	*DL/I DOS/VS Guide for New Users*
SH24-5002	*DL/I DOS/VS Diagnostic Guide*
SH24-5009	*DL/I DOS/VS Application Programming: High Level Programming Interface*
SH24-5011	*DL/I DOS/VS Data Base Administration*
SH24-5021	*DL/I DOS/VS Resource Definition and Utilities*
SH24-5022	*DL/I DOS/VS Application and Data Base Design*
SH24-5029	*DL/I DOS/VS Interactive Resource Definition and Utilities*
SH24-5030	*DL/I DOS/VS Recovery/Restart Guide*
SX24-5103	*DL/I DOS/VS Reference Summary - CALL and RQDLI Interfaces*
SX24-5104	*DL/I DOS/VS Reference Summary - System Programming*
SX24-5120	*DL/I DOS/VS Reference Summary - High Level Programming Interface*

Users of DL/I DOS/VS releases prior to Version 1.6 will recognize that the library has been restructured. Several new books have replaced previously existing books and one book, *DL/I DOS/VS General Information*, was reorganized. "Appendix A. Publication Cross-reference" on page A-1 will assist you in locating information in the restructured library that you were accustomed to finding in certain books under the library structure that existed prior to Version 1.6.

The purpose of this book is to assist you in locating information in the DL/I DOS/VS Library.

One of the traditional ways of guiding a reader to a book is to associate the book's title with a particular job title or user-role; for example, "System Programmer's Guide." However, because data processing installation organizations differ widely, this association may not always be appropriate. Another way is to associate a book, or serveral books, with a basic task essential to the use of a program in a data processing installation. With this approach, a given book can relate to a task (or subtask -- some portion of a basic task) without concern for the job title of the person responsible for doing the task. This is the approach taken with the DL/I DOS/VS Library and in organizing the material presented in Section 1 of this book.

Definitions of each of the basic tasks and each user-role follow the description on how to use this guide to the DL/I DOS/VS Library.

How To Use This Book

This book is divided into two sections:

- "Section 1. The DL/I DOS/VS Library" looks at the DL/I books from two perspectives to give you an idea of which ones may best satisfy your information needs:

  - according to a specific task that you may perform, e.g., administration, application development, operation, etc.

  - according to user-role, e.g., DP manager, system programmer, etc.

  Included with the description of each DL/I DOS/VS book is a list of other DL/I books that are considered prerequisite reading for a full understanding of the book being described.

  You will find the first section especially helpful if you are just becoming acquainted with DL/I for the first time. If you are already familiar with prior releases of DL/I, you will find the first section helpful for orientation to the new library organization.

- "Section 2. Master Index" consolidates the indexes of the other DL/I reference manuals to assist you in locating the book or books that contain information on a specific topic. It lists topics alphabetically and refers you to books in which a given topic is discussed. The index of the individual book will show you the page or pages on which the information is found.

  Note: The Master Index does not include topics from either of the DL/I DOS/VS Logic Manual volumes.

A glossary is included to help you become familiar with DL/I terminology.

General Definition of User Tasks

The following set of descriptions define the basic tasks that relate to using an IBM program. In alphabetic order they are:

Administration — Making fundamental decisions about procedures that are followed during the implementation tasks of installation, customization, operation, resource definition, application programming, and program service.

Administration is an iterative task in that many of the decisions are made before installation, continually evaluated after installation, and revised as appropriate.

Application Programming — Designing, coding, compiling, executing, debugging, and testing application programs.

Customization — Enhancing or extending an IBM program by using services and built-in facilities provided by IBM for enhancement and extension of its programs. Customization activities include coding of SVC routines and exit routines, and use of restricted macros and commands.

End Use — Directly using an IBM program to achieve the purpose for which it is provided.[1]

Evaluation — Judging the applicability of an IBM program to an installation. This task ends in a decision to install the program or not to install it.

Installation — Making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it, as required.

Operation — Starting and stopping programs, checking and controlling programs, recording the status of programs and data, and reacting to abnormal events, and recovering DL/I DOS/VS in the event of a system failure.

Program Service — Describing, resolving, and correcting an IBM program problem. Program service encompasses collecting and examining problem related facts, organizing significant facts into a problem

---

1 Because DL/I DOS/VS is not used *directly*; that is, user-supplied application programs are needed to process the data managed by DL/I for presentation to end-users, End Use information is not provided in the DL/I DOS/VS Library. In its place, information is provided for other specific user tasks; for example, Application Programming.

	description, isolating the cause of the problem, developing a fix for the problem, and testing the fix for the problem.
Resource Definition	Defining the characteristics of data processing resources to an IBM program. These resources are processor cycles, real and virtual storage, networks, terminals, input/output paths, data bases, data files, programs, user profiles, and queues.

Definition of User Roles

People who need a particular publication are listed as the intended audience of the publication. The audience is categorized by the way a person would use DL/I DOS/VS; that is, by user-role. These user-roles do not necessarily identify job titles within an organization; they are used only to identify job functions related to DL/I.

Application Designer	Responsible for detailed design and specification of a data processing system, including specification of data structures, to meet the objectives and requirements of an installation.
Application Programmer	Designs program logic to process data according to specifications; codes programs; tests, debugs, and prepares programs for execution.
Data Base Administrator	Responsible (full or part time) for defining, maintaining, and controlling the use of the installation's data base.
Data Processing Manager	Administrative and technical head of all data processing activities of the installation.
System Operator	Uses the system console to initiate single jobs, batch job processing, multiprogramming, and to handle high priority job requests not initially in the planned work load; for example, data base recovery.
System Programmer	Responsible for implementation and maintenance of programming systems, including data base management systems, to meet the needs of the installation.

Contents

Section 1. The DL/I DOS/VS Library

This section describes the books and other types of information provided for DL/I DOS/VS. The descriptions are arranged according to the tasks to which the information primarily relates:

Evaluation

Administration

Installation

Customization

Resource Definition

Application Programming

Operation

Program Service

In addition to the sources of information that relate specifically to these tasks, there are books that comprehensively apply to several tasks. These are described under "Cross-Task Manuals" on page 1-14.

Evaluation

DL/I DOS/VS General Information, GH20-1246

Intended Audience: Data Processing Management

Prerequisite DL/I DOS/VS Publications: None

Description: The General Information manual presents an introductory description of Data Language/I DOS/VS and how it may apply to the data processing needs of an installation.

The publication contains:

- An introduction to the data base concept, citing its advantages and the part it plays in a data system.
- A discussion of DL/I DOS/VS highlights, covering:
  - data management; including data redundancy, data security, and data recovery
  - productivity; the impact on programmer productivity in relation to data and device independence
  - operating environments; online, batch, and MPS batch
- A description of the characteristics of DL/I, including hierarchical data structures, data organizations, and access methods.
- A discussion of how DL/I might be put to use in various types of industries such as, manufacturing, financial, health, and processing.
- A description of possible system configurations.
- A summary of user responsibilities.

Additional information: Additional information for evaluating DL/I DOS/VS on a more technical basis may be obtained from *DL/I DOS/VS Application and Data Base Design*. That book describes in detail the kind of data base processing functions DL/I provides.

Administration

DL/I DOS/VS Data Base Administration, SH24-5011

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications: None

Description: This book is a comprehensive guide for data base administration activities pertaining to DL/I DOS/VS. It describes what is involved in planning application installation, implementing the data base design, and controlling a DL/I data base management system. By providing detailed discussions of all necessary concepts, tasks, and considerations, this book leads you through making appropriate decisions for using the generation, maintenance, and recovery utility programs described in the *Resource Definition and Utilities* manual or for using the Interactive Macro Facility to define data bases and their use by application programs.

The four main sections of the book discuss:

- Installation Planning -- covers pre-installation planning for new applications and the subsequent installation of additional applications.
- Implementing the Data Base Design -- covers the preparation and implementation steps for creating a physical data base for use in a particular operating environment. The process of developing the data base design is described in *DL/I DOS/VS Application and Data Base Design.*
- Using Data Bases -- covers application program execution, data base loading, and how to modify an existing data base.
- Controlling Data Base Operation -- covers data security options, performance monitoring and modification, and data base recovery planning.

Additional Information: Chapter 7 in *DL/I DOS/VS Application and Data Base Design* describes the considerations and decisions involved in physically implementing the system view of the application. It is a link between the tasks of application design and data base administration.

Installation

Installation of a DL/I DOS/VS data base management system involves more than the physical installation of the program product itself. It requires decisions on matters concerning data base design, data organizations and access methods, use of optional functions, the use of data by application programs, etc. These are all considerations that can be classified with the more general task of administration. Therefore, they are discussed in *DL/I DOS/VS Data Base Administration.*

Information relating to physically installing DL/I DOS/VS on VSE is provided in the Program Installation Directory supplied with the program materials. It describes the specific requirements and procedures for installing the program.

Customization

Customization of DL/I DOS/VS is achieved through the regularly supplied DL/I macros, user-selectable optional functions, and user exits. Decisions regarding which facilities to use and how to use them are considerations associated with the general task of administration. Therefore, they are discussed in *DL/I DOS/VS Data Base Administration.*

Resource Definition

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

Description: This manual provides procedural and reference information for implementing the tasks described in the *DL/I DOS/VS Data Base Administration* book. It describes procedures for:

- creating control blocks to define resources to DL/I
- using utilities for resolution of data base logical relationships and secondary indexes, data base reorganization, and data base recovery

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

Description: This manual contains introductory and procedural information about the DL/I DOS/VS Interactive Macro Facility (IMF) and Interactive Utility Generation (IUG) facility. IMF and IUG offer easy-to-use interactive procedures that let you perform resource definition and utility functions at a terminal.

IMF allows you to generate jobs to create, modify, and delete DL/I control blocks. The IMF portion of this manual describes how to create:

- A data base description (DBD)
- A program specification block (PSB)
- An application control table (ACT).

It also describes how to migrate ELIAS-defined DBD and PSB definition tables to IMF format.

IUG lets you generate job streams for each of the DL/I utilities. The IUG portion of this manual describes how to generate job streams to:

- Reorganize a data base
- Load a data base
- Recover a data base
- Assist in problem determination

- Extract data from a DL/I data base to put in ISQL format.

This section on IUG also describes how to resume an interrupted job stream generation dialog and how to update and view label and table information.

As an aid in preparing for a terminal session, several series of worksheets corresponding to actual data entry panels are provided. Worksheets for defining a data base description (DBD), a program specification block (PSB), and an application control table (ACT) are included.

DL/I DOS/VS IMF and IUG Online Information

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

Description: Online information consists of display terminal panels (display images) that supply introductory and reference information for using the DL/I Interactive Macro Facility and Interactive Utility Generation facility. The display panels consist of:

- menus offering a choice of the types of online information available for viewing at the terminal:
  - An overview of IMF and IUG
  - Using DL/I interactive macros for defining, modifying, or deleting DBDs, PSBs, and ACTs; and for migrating ELIAS-I (Entry Level Interactive Application System - One) tables to IMF
  - Using the DL/I IUG facility to generate the job streams for each of the DL/I utilities
  - An explanation of the DL/I primary menu selections
  - A list of DL/I reference publications
  - A table explaining the WHAT NEXT and NEXT MACRO entries that appear at the end of most IMF data entry panels

  Each choice is supplemented by HELP panels containing additional information.
- HELP panels which supplement each data entry panel by supplying sufficient reference information to enable correct completion of the corresponding data entry panel. In cases where additional reference information exists in a printed DL/I manual, the HELP panel points to the appropriate manual.

DL/I DOS/VS Reference Summary - System Programming, SX24-5104

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH12-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

Description: This publication is a reference card providing quick synoptic reference information relating to DL/I DOS/VS. Examples of the kinds of information included are: job control statements, DBD and PSB generation statements, etc.

Application Programming

DL/I DOS/VS Application and Data Base Design, SH24-5022

Intended Audience: Application Designer, Data Base Administrator

Prerequisite DL/I DOS/VS Publications: None

Description: This book describes a method of performing application design when DL/I DOS/VS and DL/I data bases are used. It is organized to lead you step-by-step through the process of application design for applications using DL/I.

The three main parts of this book are:

- Application Design -- covers collecting and analyzing application requirements, and relating them to data base design.
- Data Base Design -- covers creating views of data for individual applications and resolving them into a system-wide physical view.
- Implementing the Application -- covers suggestions for developing and carrying out a plan for implementing the application design.

DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009

Intended Audience: Application Programmer

Note: Application programmers using the CALL interface to DL/I for COBOL, PL/I, or Assembler languages, or RQDLI for RPG II, do not need this book; but should use *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces* instead.

Prerequisite DL/I DOS/VS Publications: None

Description: This book provides information needed for planning, writing, debugging, and executing data base application programs. It is directed to programmers using the DL/I High Level Programming Interface (HLPI) in conjunction with the COBOL or PL/I Optimizer language.

The book contains:

- An overview of DL/I and data bases for the new DL/I application programmer.
- A general introduction to DL/I High Level Programming Interface and the way in which it is used.
- A description of High Level Programming Interface syntax.
- A summary of the process of planning, writing, and debugging an application program using High Level Programming Interface.

- A reference section listing each of the HLPI commands, with the information needed to code and use them.

DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411

Intended Audience: Application Programmer

Note: Application programmers using the DL/I High Level Programming Interface do not need this book; but should use *DL/I DOS/VS Application Programming: High Level Programming Interface* instead.

Prerequisite DL/I DOS/VS Publications: None

Description: This book provides information needed for planning, writing, debugging, and executing data base application programs. It is directed to programmers using the DL/I CALL interface in conjunction with the COBOL, PL/I, or Assembler language, or the RQDLI interface for RPG II.

The book contains:

- An overview of DL/I and data base concepts for the new DL/I application programmer.
- A summary of the process of planning, writing, and debugging an application program using either the CALL or RQDLI interface.
- A reference section showing each of the call functions, with the information needed to code and use them.
- A discussion of considerations unique to online application programs.
- A summary of the use advanced functions; for example, command codes, secondary indexing, etc.

Low-Level Code/Continuity Check in Data Language/I DOS/VS Program Reference and Operations Manual, SH20-9046

Intended Audience: Application Designer, Application Programmer, Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Application and Data Base Design, SH24-5022

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411

Description: This manual is intended for those developing and implementing application programs in the manufacturing industry. It describes the functions and facilities of Low-Level Code/Continuity Check in DL/I DOS/VS.

The manual contains:

- An introduction and general description which defines problem areas in ensuring assembly-to-subassembly continuity within the manufacturing industry environment and the solutions offered by Low-Level Codes/Continuity Check under DL/I.

- A description of the structure and functions of Low-Level Codes/Continuity Check and its data base, including:

  - data base description
  - invocation of Low-Level Codes/Continuity Check
  - operational procedures
  - error messages and return codes

- A discussion of installation requirements for Low-Level Codes/Continuity Check.

DL/I DOS/VS Reference Summary - High Level Programming Interface, SX24-5120

Intended Audience: Application Programmer

Note: Application programmers using the CALL interface to DL/I for COBOL, PL/I, or Assembler languages, or RQDLI for RPG II, do not use this reference card; but should use *DL/I DOS/VS Reference Summary - CALL and RQDLI Interfaces* instead.

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009

Description: This publication is a reference card providing HLPI application programmers quick synoptic reference information relating to DL/I DOS/VS. Examples of the kinds of information included are: status codes, HLPI command formats and parameters, etc.

DL/I DOS/VS Reference Summary - CALL and RQDLI Interfaces, SX24-5103

Intended Audience: Application Programmer

Note: Application programmers using the DL/I High Level Programming Interface do not use this reference card; but should use *DL/I DOS/VS Reference Summary - High Level Programming Interface* instead.

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411

Description: This publication is a reference card providing application programmers who use the standard CALL interface or the RQDLI interface quick synoptic reference information relating to DL/I DOS/VS. Examples of the kinds of information included are: standard CALL statement formats and parameters, SSA usage, status codes, etc.

Operation

DL/I DOS/VS runs as an application program in a VSE system. Because of this, and due to widely varying operational procedures among VSE system installations, specific operational information is not provided. However, inasmuch as system operation may involve activities related to data base reorganization or data base recovery, information pertaining to running utility programs to accomplish these activities is found in:

- *DL/I DOS/VS Resource Definition and Utilities*
- *DL/I DOS/VS Interactive Resource Definition and Utilities*
- *DL/I DOS/VS Recovery/Restart Guide*

System operators should also have access to the information in *DL/I DOS/VS Messages and Codes*, which is described under "Cross-Task Manuals" on page 1-14.

DL/I DOS/VS Recovery/Restart Guide, SH24-5030

Intended Audience: Data Base Administrator, System Programmer

Prerequisite DL/I DOS/VS Publications

- *DL/I DOS/VS Messages and Codes*, SH12-5414
- *DL/I DOS/VS Data Base Administration*, SH24-5011
- *DL/I DOS/VS Resource Definition and Utilities*, SH24-5021
- *DL/I DOS/VS Interactive Resource Definition and Utilities*, SH24-5029

Description: This book helps you to plan and develop a recovery and restart procedure in the event of a system failure. It discusses the various facilities available for recoverying and restarting your system after a failure, including a description of the steps that can be used for a normal recovery procedure. Since the DL/I utilities are used to recover and restart your system, this manual also provides a summary of the job control statements necessary to run them.

The main parts of this book are:

- Planning -- covers the planning considerations for developing recovery/restart capabilities.
- Facilities -- covers the facilities that can be used for planning and effecting recovery and restart.
- Recovery Guide -- covers the steps that can be used in normal recovery and possible actions for recovering from a failure that occurs during the recovery process itself.
- JCL Summary -- provides a summary of the job control statements necessary to run the DL/I utilities you may use in your recovery/restart process.

Program Service

DL/I DOS/VS Diagnostic Guide, SH24-5002

Intended Audience: System Programmer, Data Base Administrator

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

DL/I DOS/VS Messages and Codes, SH12-5414

Corequisite Publications

DL/I DOS/VS Logic, Volume 1, LY12-5016

DL/I DOS/VS Logic, Volume 2, LY24-5215

Description: The purpose of this publication is to assist in analyzing and isolating DL/I production or testing problems. It is intended for installation personnel responsible for system programming and system maintenance.

This publication contains:

- A high-level general review of DL/I job control statements, control blocks, calls and commands, data base organization, and access methods.
- A discussion of DL/I control flow, scheduling errors, online trace, logical retrieve, and online wait state debugging, including suggestions to assist in isolating problems.
- A description of the types of dumps DL/I produces, with suggestions of possible paths to follow in finding the cause of the dump through analysis of selected messages.
- A discussion of problems associated with various request handlers, initialization/termination, utility, and test programs. Included are major causes for program failure, things that must be done when using these programs, and suggestions for determining the cause of a program failure.
- A description of the Interactive Macro Facility (IMF), including module flow diagrams, table information, and an example of a VSE system job stream for generating a DBD control block.
- A description of the Interactive Utility Generation (IUG) facility, including module flow diagrams, table information, and examples of a VSE system job stream as generated by IUG for each DL/I utility.
- A discussion of the use of the DL/I Trace Facility, including its operands and parameters, in problem determination.

- A description of the Trace Print Utility for selectively printing the output of the DL/I Trace Facility.

DL/I DOS/VS Logic, Volume 1, LY12-5016

DL/I DOS/VS Logic, Volume 2, LY24-5215

Intended Audience: System Programming, Data Base Administrator

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

DL/I DOS/VS Messages and Codes, SH12-5414

Corequisite Publications

DL/I DOS/VS Diagnostic Guide, SH24-5002

Description: These manuals provide information on the internal design and function of DL/I DOS/VS. They are intended for use, together with the program listings for DL/I DOS/VS, by persons engaged in program maintenance and modification.

Volume 1 contains:

- A summary of the purpose of system control modules, DL/I facility modules, MPS modules, and utility modules.
- A description of DL/I modules and major routines.
- A directory that lists DL/I module, entry point, and control section names with cross-references to HIPO diagrams contained in Volume 2.
- A description of data areas used by DL/I, including field and flag names for each data area.
- A list of DL/I messages and status codes cross-referencing the modules that originate them.

Volume 2 contains the Method of Operation (HIPO) diagrams that describe the DL/I modules and provide cross-references to labels in the program listings.

Cross-Task Manuals

DL/I DOS/VS Messages and Codes, SH12-5414

Intended Audience: Application Programmer, System Operator, System Programmer, Data Base Administrator

Prerequisite DL/I DOS/VS Publications

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009

DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411

Description: This manual provides reference information for all DL/I messages and status codes (except those presented at the terminal by the Interactive Macro Facility), including an explanation of the possible causes for the message or code and suggested responses.

The manual contains:

- A description of each message issued by DL/I DOS/VS during execution.
- A description of messages (MNOTES) issued during:
  - data base description (DBD) control block generation
  - program specification block (PSB) control block generation
  - online nucleus generation
- A description of messages and return codes issued by the DL/I DOS/VS Low-Level Code/Continuity Check application support program.
- A description of the status codes returned following a DL/I command or call request from an application program.

DL/I DOS/VS Guide for New Users, SH24-5001

Intended Audience: Data Base Administrator, Application Designer, Application Programmer, System Programmer

Corequisite Publications

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029

DL/I DOS/VS Application and Data Base Design, SH24-5022

DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009

DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411

Description: This publication gives new users of DL/I a foundation for planning, designing, and building a data base system. It provides information needed to initially install and build a DL/I data base system. In addition, the publication provides information to assist in developing application programs.

This manual describes the sample application supplied by IBM and discusses, through the use of examples, how the sample application was designed and implemented. It provides workable examples for setting up a specific data base application and serves as a guide for writing data base application programs.

Section 2. Master Index

This master index consolidates the indexes of the books in the DL/I DOS/VS library referenced most frequently in the course of day-to-day use of DL/I. It does not contain entries from the indexes of the logic manuals which are primarily used for program service.

Entries in the master index refer you to the DL/I manuals through use of an uppercase alphabetic code which represents the manual title. For example, ADBD represents the *DL/I DOS/VS Application and Data Base Design* book. A foldout at the back of the book lists all of the manual codes, together with the corresponding manual title and order number. When opened, the foldout extends beyond the right-hand margin of the book for convenient use along side the master index.

The following alphabetic codes are used in the master index to identify specific DL/I DOS/VS books.

ADBD *DL/I DOS/VS Application and Data Base Design,* SH24-5022

CALL *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces,* SH12-5411

DBA *DL/I DOS/VS Data Base Administration,* SH24-5011

DIAG *DL/I DOS/VS Diagnostic Guide,* SH24-5002

GIM *DL/I DOS/VS General Information,* GH20-1246

GNU *DL/I DOS/VS Guide for New Users,* SH24-5001

HLPI *DL/I DOS/VS Application Programming: High Level Programming Interface,* SH24-5009

IRDU *DL/I DOS/VS Interactive Resource Definition and Utilities,* SH24-5029

LLC *Low-Level Codes/Continuity Check in Data Language/I DOS/VS, Program Reference and Operations Manual,* SH20-9046

RDU *DL/I DOS/VS Resource Definition and Utilities,* SH24-5021

RRG *DL/I DOS/VS Recovery/Restart Guide,* SH24-5030

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

Q

R

S

U

V

W

X

0

Appendix A. Publication Cross-reference

This appendix provides cross-references to information on topics covered in books replaced or restructured for DL/I DOS/VS Version 1, Release 6.

Cross-reference is accomplished through use of an alphabetic code that represents the title of the manual in which similar information, *at a similar level of detail*, is found. You should bear in mind that information on the same topic, at a different level of detail, might be found in books other than those shown by the cross-reference code. The Master Index will help you in locating the additional sources.

Note: Information formerly contained in the *DL/I DOS/VS IMF User's Guide*, SH24-5007, has been moved to a new book published for DL/I DOS/VS Version 1, Release 7. The new book is *DL/I DOS/VS Interactive Resource Definition and Utilities*, SH24-5029. No cross-referencing is shown in this appendix because the transfer of information was made in its entirety from one book to the other to produce a new document for the interactive environment.

How to Use This Appendix

Cross-referencing is based on the Table-of-Contents of the Version 1.5 books that have been replaced or restructured.

If the content of an entire chapter, appendix, etc., from a Version 1.5 book has moved so that corresponding information is contained without exception in a single book, only the heading of that chapter, etc., shows the cross-reference code.

On the other hand, if the content of a chapter, etc., has moved so that corresponding information is not contained in a single book, each individual entry is cross-referenced.

Note: In most cases the wording of the table-of-contents entries in the new books is not the same as the original entry; therefore, using the index of the new book will give you quicker access to the information.

A foldout at the back of the book lists all of the manual codes, together with the corresponding manual title and order number. When opened, the foldout extends beyond the right-hand margin of the book for convenient use alongside this appendix.

The following alphabetic codes are used in this appendix to identify specific DL/I DOS/VS books.

ADBD *DL/I DOS/VS Application and Data Base Design,* SH24-5022

DBA *DL/I DOS/VS Data Base Administration,* SH24-5011

GIM *DL/I DOS/VS General Information,* GH20-1246

LIB *DL/I DOS/VS Library Guide and Master Index,* GH24-5008

RDU *DL/I DOS/VS Resource Definition and Utilities,* SH24-5021

The code **N/D** (meaning "not documented") indicates that the information contained under the entry is no longer presented in the context it was in the

Version 1.5 book. For instance, the N/D code applies to such things as the brief descriptions of functional enhancements supplied by previous releases -- the information is now properly intergrated in the library -- and to introductory material that is no longer appropriate in the new DL/I books.

Where an alternate source of information relating to a topic coded N/D exists, a footnote is used to identify that source. However, the information in the alternate source may be significantly different in its level of detail.

DL/I DOS/VS General Information

Chapter 1: Introduction
- Why Data Bases? GIM
- What Is a Data Base? GIM
- What Does a Data Base Provide GIM
- How is the DL/I Data Base Implemented GIM, DBA

Chapter 2: General Description of DL/I
- DL/I Batch System GIM, DBA
  - Initialization Module DBA
  - Language Interface Module DBA
  - Program Request Handler DBA
  - DL/I Facility DBA
- DL/I Online Processor GIM, DBA
- Multiple Partition Support (MPS) GIM, DBA
- Utility Programs and Procedures GIM, DBA
  - Program Specification Block (PSB) Generation GIM, DBA
  - Data Base Description (DBD) Generation GIM, DBA
  - Application Control Blocks Creation and Maintenance Utility DBA
  - Data Base Reorganization Unload and Reload GIM, DBA
  - Data Base Recovery GIM, DBA
- Application Support Program GIM
  - Low-Level Code and Continuity Check GIM

Chapter 3: DL/I System Concepts
- DL/I Data Base Structure GIM, ADBD
- Hierarchical Data Structure GIM
  - Definitions GIM, ADBD
  - Logical Data Structures/Physical Data Structures GIM, ADBD
  - Sequence Fields and Access Paths ADBD
  - Data Base Definition GIM
  - DL/I, the Application Program Interface DBA
  - Logical Relationships ADBD
  - Data Base User Interface N/D[1]
  - Data Base Organization and Access Methods GIM
  - Segment Definition and Format ADBD
  - Interrelated Data Base Records ADBD
  - Indexed and Indexing Data Base Records ADBD
- Data Base Administration GIM, DBA
- Online Processing Capability GIM, DBA

Chapter 4: User Installation Requirements
- User Installation Responsibilities GIM, ADBD
- User Schedule DBA

Chapter 5: Machine Configurations GIM
- Minimum DL/I Configuration
- Typical DL/I Real Storage Requirements

---

1 See either *DL/I DOS/VS Application Programming: High Level Programming Interface* or *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces* for information concerning application program interfaces to DL/I DOS/VS.

Chapter 6: Programming Requirements
- For DL/I Version 1.4 N/D[2]
- For DL/I Version 1.4 ICR N/D[2]
- For DL/I Version 1.5 N/D[2]
- For DL/I Version 1.5 ICR N/D[2]

Chapter 7: Sample Applications GIM
- Manufacturing Industry
- Financial Industry
- Medical Industry
- Process Industry
- The Data Dictionary

Appendix A: Comparison to IMS/VS and DL/I-Entry DOS/VS N/D[3]

Appendix B: DL/I DOS/VS Enhancements N/D
- DL/I Version 1.2
- DL/I Version 1.3
- DL/I Version 1.4
  - RPG II Support
  - Prefix Resolution Improvement
  - Trace Modification
  - Extended DL/I Call Interface
  - Intersystem Communication
  - High Level Language Debugging for PL/I
  - Performance Improvements
  - Program Isolation Enhancements
  - DL/I in the CMS/DOS Environment
- DL/I Version 1.4 ICR
  - Fixed Block Architecture Support
  - VSE/ICCF Support
- DL/I Version 1.5
  - Field Level Sensitivity
  - Extended Logical Relationships
  - Unique Segment Support
  - All-Partition MPS Support
  - IPCS Dump Hook Support
- DL/I Version 1.5 ICR
  - DL/I High Level Programming Interface
  - ISC CHECKPOINT Support

---

2 See *DL/I DOS/VS Data Base Administration* for equivalent information.

3 Comparison between DL/I DOS/VS and DL/I-Entry is no longer documented. See *DL/I DOS/VS Data Base Administration* for information on DL/I DOS/VS compatibility with IMS/VS.

DL/I DOS/VS System/Application Design Guide

Chapter 1: Planning the Implementation of DL/I
- DL/I Environment GIM
- Personnel Environment DBA
  - Application Design ADBD
  - Data Base Design ADBD
  - System Design DBA

Chapter 2: Designing Data Base Structures ADBD
- Data Base Record
- Hierarchical Data Base Structure
  - Rules for Data Base Structures
- Segment Design
- Hierarchy Design
- Completing the Design

Chapter 3: Execution and Control of DL/I Data Base System
- System Installation Considerations DBA
  - DL/I System Modules DBA
  - DL/I Access Method Requirements DBA
- Requirements for Execution DBA
  - Data Base Description (DBD) Generation DBA
  - Program Specification Block (PSB) Generation DBA
  - DL/I Application Control Blocks Creation and Maintenance DBA
  - Application Program DBA
  - Data Base Space Allocation DBA
- DL/I Batch System Execution DBA
  - DL/I Batch System Control Flow DBA
- DL/I Online System Execution DBA
  - DL/I Online System Control Flow DBA
  - MPS Batch System Flow DBA
- Intersystem Communication Support DBA
- Data Base Buffering DBA, RDU
- Data Base Logging and Recovery DBA
  - Asynchronous Logging Option DBA
  - Logging and Performance DBA
  - Choosing the DL/I Log Medium DBA
- DL/I Disk Logging Consideration DBA
- DL/I Checkpoint DBA
  - Related File Processing Considerations DBA
  - Application Program Restart Considerations DBA
  - DL/I Checkpoint in Batch MPS Programs DBA
  - DL/I Checkpoint in Online Transactions DBA

Chapter 4: Application Program Design
- Batch Application Program Design DBA
  - General Considerations DBA
  - Programming Language to be Used N/D
  - Establishing Useful Conventions N/D
  - Testing DBA
  - Naming Conventions ADBD
  - Use of Source Statement Libraries N/D
  - Use of a DOS/VS Procedure Library N/D

Application Program Structure N/D[4]
Job Structure and Program Libraries N/D
Status Code Checking N/D[4]
Positive Segment Processing N/D[4]
Online Application Program Design DBA
General Considerations DBA
Differences Between Batch, MPS, and Online DL/I DBA
Conventions N/D[4]
Reentrant Coding for DL/I N/D[4]
Use of DOS/VS Communication Region N/D[4]
Intent Scheduling vs Program Isolation DBA
DL/I Scheduling DBA
Program Isolation DBA
Debugging N/D[4]
Status Code Analysis N/D[4]
Tracing N/D[5]
MPS Application Program Design DBA

Chapter 5: Data Base Design and Access Method Selection

Concepts of Physical Data Bases DBA
Segments DBA
Segment Formats DBA
Fields DBA
Structure DBA
Data Base Calls N/D[4]
Special DL/I Calls DBA
Data Base Organization and Access Methods DBA
Hierarchical Sequential -- Definition DBA
Hierarchical Direct -- Definition DBA
Logical Relationships DBA
Unidirectional Logical Relationships DBA
Bidirectional Logical Relationships DBA
Logical Relationship Paths DBA
Logical Child Segment DBA
Example of Unidirectional Logical Relationships DBA
Example of Bidirectional Logical Relationships DBA
Logical Child Sequence Fields DBA
Pointers in Logical Relationships DBA
Rules for Defining Logical Relationships in Physical
Rules for Insertion, Deletion, and Replacement DBA
Defining a Logical Data Base DBA
Secondary Indexing DBA
Secondary Processing Sequence DBA
Secondary Data Structure DBA
Index Pointer Segment Format DBA
XDFLD Statement DBA
System Related Fields DBA
Additional Data in Index Pointer Segments DBA
Suppression of Index Entries DBA
Processing With Secondary Indexing DBA
Using the Secondary Processing Sequence DBA
Processing a Secondary Index as a Data Base DBA

---

4 See either *DL/I DOS/VS Application Programming: High Level Programming Interface* or *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces* for detailed information concerning this topic.

Variable Length Segments DBA
Segment Edit/Compression Exit DBA
Field Level Sensitivity DBA
Virtual Fields DBA
Automatic Data Format Conversion DBA
User Field Exit Routine DBA
Dynamic Segment Expansion DBA
Additional Field Sensitivity Considerations DBA
Application View of Search Fields for Alternate Processing Sequence DBA
Logical Record and Block Formats N/D[5]
HSAM Logical Record Block Formats N/D[5]
HISAM Logical Record Block Formats N/D[5]
Hierarchical Direct - Logical Record and Block Formats N/D[5]
HDAM and HIDAM Storage Organizations DBA
HDAM DBA
HIDAM DBA
Format of Files Used for HDAM and HIDAM N/D[5]
Comparison of HIDAM Load and HDAM Load DBA
Hierarchical Direct -- Insertion and Deletion N/D
Data Base Design Considerations DBA
Hierarchical Sequential Design Considerations DBA
Hierarchical Direct Design Considerations DBA
Viability of Data Base Design DBA

Chapter 6: Utilities DBA
Data Base Recovery
Data Base Reorganization
Reorganization Interval
Reorganization of HISAM Data Bases
Reorganization of HDAM and HIDAM Data Bases
Logical Relationship Resolution

Chapter 7: DL/I Virtual Storage Estimates DBA
DL/I Batch System Storage Requirements
DL/I Initialization and Termination Routines
DL/I Nucleus and Program Request Handler
DL/I Program Specification Block (PSB)
DL/I Data Management Blocks (DMB) Derived from
Data Base Descriptions (DBD)
DL/I Data Base Buffer Pool and Control Blocks
DL/I Modules--Data Base Organization Dependent
DOS/VS Modules--Access Method Dependent
DOS/VS Buffer Pools and Control Blocks
DL/I Online System Storage Requirements
DL/I Online Nucleus
DL/I Tables (ACT, SCD, PPST, PDIR, RPDIR, DDIR)
DL/I Partition Specification Table (PST)
Additional PSB Storage
DL/I MPS System Storage Requirements
DL/I Batch System Storage Requirement Example
DL/I Data Base Utilities Storage Requirements
Application Control Blocks Creation and Maintenance
Data Base Data Set Image Copy Utility - DLZUDMP0

---

5 See *DL/I DOS/VS Diagnostic Guide.*

DL/I DOS/VS Utilities and Guide for the System Programmer

Chapter 1: Introduction
- Coding Conventions RDU
- Using the Interactive Macro Facility N/D[6]
- Implementation Overview DBA

Chapter 2: Data Base Description (DBD) Generation RDU
- How to Create a DBD Interactively
- Types of DBD Generation
  - Simple HSAM DBD Generation
  - HSAM DBD Generation
  - Simple HISAM DBD Generation
  - HISAM DBD Generation
  - HDAM DBD Generation
  - HIDAM DBD Generation
  - INDEX DBD Generation
  - Secondary Index DBD Generation
  - Logical DBD Generation
- General Data Base Description Rules
- Physical DBD Control Statement Formats
  - DBD Statement
  - DATASET Statement
  - SEGM Statement
  - LCHILD Statement
  - FIELD Statement
  - XDFLD Statement
  - DBDGEN Statement
  - FINISH Statement
  - END Statement
- Logical DBD Control Statement Formats
  - DBD Statement
  - DATASET Statement
  - SEGM Statement
  - DBDGEN Statement
  - FINISH Statement
  - END Statement
- Execution of DBD Generation - Job Control Language
- Description of DBD Generation Output
  - Quick Check of DBD Generation
  - Load Module
- Physical Data Base Description Examples
  - HSAM DBD Generation
  - HISAM DBD Generation
  - HDAM DBD Generation
  - HIDAM DBD Generation
  - Summary of Physical Data Base Description Examples
- Logical Relationships - Physical DBD Generations Examples
  - Unidirectional Logical Relationships
  - Virtually Paired Bidirectional Logical Relationship
- Logical DBD Generation Examples

---

6 See *DL/I DOS/VS IMF User's Guide* for a complete description for Version 1 Release 6 or *DL/I DOS/VS Interactive Resource Definition and Utilities* for Version 1 Release 7.

Summary of Logical Data Base Description Examples
Secondary Index DBD Generation Example

Chapter 3: Program Specification Block (PSB) Generation RDU
PSB Rules
How to Create a PSB Interactively
PSB Control Statement Format
PCB Statement
SENSEG Statement
SENFLD Statement
VIRFLD Statement
PSBGEN Statement
END Statement
Execution of PSB Generation - Job Control Language
Description of PSB Generation Output
PSB Generation Examples
Physical Data Bases
Logical Data Bases
Data Bases with Secondary Indexes

Chapter 4: Application Control Blocks Creation and Maintenance RDU
Job Control Language Requirements
Control Statement Requirements
Examples

Chapter 5: Data Base Recovery
STXIT AB Processing in Batch (non-MPS) DBA
System/370 Power Failure DBA
DL/I System Log DBA
Online System Logging DBA
Asynchronous Logging DBA
Multivolume System Log Tapes DBA
Multifile System Log Tapes DBA
Disk Logging DBA
DL/I System Log DSECT N/D[7]
Data Base Data Set Image Copy Utility RDU
Job Control Language Requirements RDU
Control Statement Requirements RDU
Examples RDU
Data Base Change Accumulation Utility RDU
Job Control Language Requirements RDU
Control Statement Requirements RDU
Examples RDU
Data Base Data Set Recovery Utility RDU
Job Control Language Requirements RDU
Control Statement Requirements RDU
Examples RDU
Log Print Utility RDU
Job Control Language Requirements RDU
Control Statement Requirements RDU
Examples RDU
Data Base Backout Utility RDU
Job Control Language Requirements RDU

---

[7] See *DL/I DOS/VS Logic Manual, Volume 1* for information on the data base logger, DLZRDBL0.

Control Statement Requirements RDU
Examples RDU

Chapter 6: Data Base Reorganization/Load Processing
Reorganization/Load Flowcharts DBA
Data Base Initial Load/Reload DBA
With Logical Relationships DBA
With Secondary Indexes DBA
Data Base Physical Reorganization RDU
HISAM Reorganization Unload Utility RDU
HISAM Reorganization Reload Utility RDU
HD Reorganization Unload Utility RDU
HD Reorganization Reload Utility RDU
Data Base Logical Relationship Resolution Utilities RDU
Data Base Prereorganization Utility RDU
Data Base Scan Utility RDU
Data Base Prefix Resolution Utility RDU
Data Base Prefix Update Utility RDU
Data Base Reorganization/Load Processing Examples RDU

Chapter 7: Additional Responsibilities of the System Programmer
DL/I Data Base Buffer Pool DBA
HDBFR pool Control DBA
VSAM Buffer Allocation DBA
DL/I Buffer Pool Statistics DBA
Storage Layout Control Option (Online System Only) RDU
Generating a Storage Layout Control Table RDU
HDAM Randomizing Modules DBA
Randomizing Module Interfaces DBA
HDAM Randomizing Module Examples DBA
Segment Compression/Expansion Routines DBA
Compression Routines Function DBA
Compression Routine Interface DBA
Segment Compression DBA
Segment Expansion DBA
Segment Compression Table DBA
Secondary Indexing Exit Routines DBA
Secondary Indexing Exit Routine Interface DBA
User Field Exit Routine DBA

Chapter 8: DL/I System Installation and Batch Initialization
Programming Systems Used for DL/I DOS/VS Version 1.4 N/D[8]
Programming Systems Used for DL/I DOS/VS Version 1.4 ICR N/D[8]
Programming Systems Used for DL/I DOS/VS Version 1.5 N/D[8]
Minimum Machine Requirements DBA
Building the DL/I System RDU
DOS/VS Supervisor Generation N/D
DOS/VSE Supervisor Generation RDU
Relinking DL/I Modules RDU
CICS/VS - DL/I Release Dependencies DBA
Defining DL/I Data Base Structures DBA
Initialization of the DL/I Batch System RDU
DOS/VS UPSI Byte Settings for DL/I Batch System RDU

---

8 See *DL/I DOS/VS Data Base Administration* for equivalent information.

DL/I Parameter Information Requirements RDU
DL/I Initialization Job Control Language Requirements RDU
DL/I MPS Batch Partition Initialization RDU
DOS/VS UPSI Byte Settings for MPS RDU
DL/I MPS Parameter Information Requirements RDU
DL/I MPS Initialization Job Control Language Requirements RDU
Executing Batch MPS Programs RDU
Dynamically Scheduling MPS or Non-MPS Execution RDU

Chapter 9: DL/I Online System
Installation N/D
Online Nucleus Generation Requirements RDU
How to Create an Online Nucleus Interactively RDU
Online Nucleus Generation Macro RDU
Establishing the Control Section for the DL/I Application Control Table RDU
Defining the Online Environment for DL/I RDU
Describing the Application Program Relationship to DL/I Data Bases RDU
Specifying a Data Base Resident on Another System RDU
Specifying Buffer Pool Control Options RDU
Specifying the End of the DL/I Application Control Table RDU
Job Control Language for Generating the Online Nucleus RDU
Description of Online Nucleus Generation Output RDU
Online Nucleus Generation Example RDU
CICS System Generation Responsibilities for DL/I DBA
CICS System Generation (DFHSG) DBA
System Initialization Table (SIT) DBA
Journal Control Table (JCT) DBA
File Control Table (FCT) DBA
Processing Program Table (PPT) DBA
Program List Table (PLT) DBA
Program Control Table (PCT) DBA
Restrictions for DL/I DOS/VS Data Base Usage N/D
Initialization of the DL/I Online System RDU
Controlling the DL/I Online System Environment DBA
DL/I System Call Format and Returns DBA
Scheduling the DL/I System Call DBA
DL/I System Call Examples DBA
CMXT Call Example DBA
STRT and STOP Call Example DBA
TSTR and TSTP Call Example DBA

Chapter 10: Multiple Partition Support (MPS)
MPS Components N/D[9]
Component Functions N/D[9]
MPS Programming Considerations DBA
MPS Requirements DBA
CICS/VS File Control Table (FCT) DBA
CICS/VS Journal Control Table (JCT) DBA
CICS/VS Program Control Table (PCT) DBA
CICS/VS Program Processing Table (PPT) DBA
CICS/VS System Initialization Table (SIT) DBA
CICS/VS Program List Table (PLT) DBA

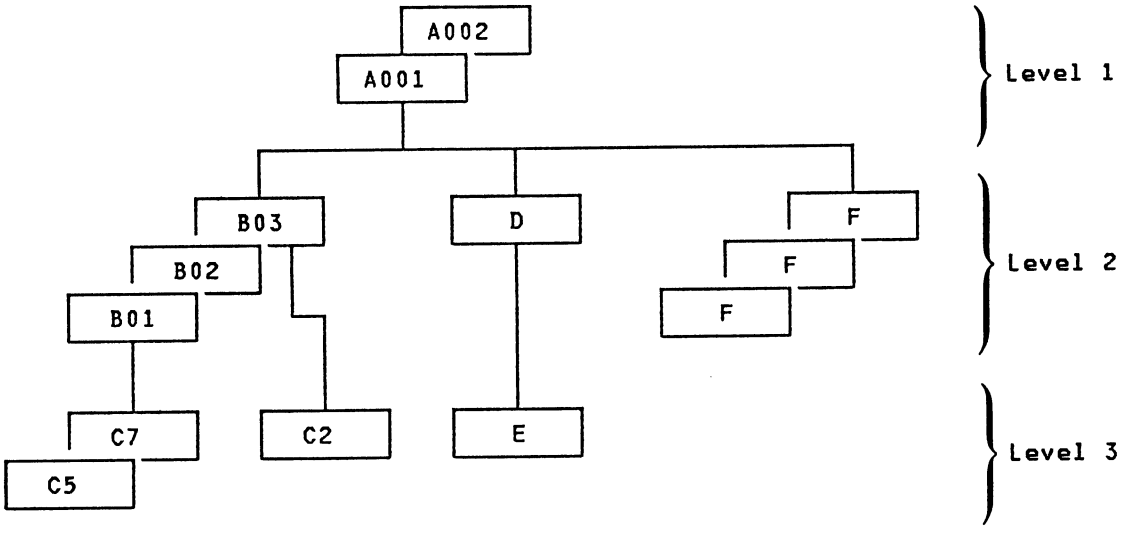
---

9 See *DL/I DOS/VS Logic Manual, Volume 1* for equivalent information on MPS components.

Glossary

A number of acronyms, terms, and phrases used in describing DL/I DOS/VS are either new to most readers, or have new meanings. To improve the readability and your understanding of this and other DL/I DOS/VS publications, the significant and important terms are defined in this glossary.

Some of the definitions refer to the representative DL/I hierarchical structure shown in Figure A-1.



- Each block represents a segment.
- The segment names are A through F.
- The numbers represent the sequence fields (keys). If no number is present, the segment has no key.
- The lines connecting the segment blocks show the hierarchical paths.

Figure A-1. Representative DL/I Hierarchical Structure

ACB. (1) Application control blocks (DL/I). (2) Access method control block (VSAM).

ACBGEN. Application control block generation.

access method control block (ACB). A control block that links a program to a VSAM data set.

access method services. A multifunction utility program that defines VSAM data sets (or files) and allocates space for them, and lists data set records and catalog entries.

ACT. Application control table.

addressed direct access. In systems with VSAM, the retrieval or storage of a data record identified by its relative byte address, independent of the record's location relative to the previously retrieved or stored record. (See also *keyed direct access, addressed sequential access, keyed sequential access*, and *relative byte address*).

addressed sequential access. The retrieval or storage of a VSAM data record relative to the previously retrieved or stored record. (See also *keyed sequential access, addressed direct access*, and *keyed direct access*).

aggregate. See *data aggregate.*

anchor point (AP). See *root anchor point.*

application control blocks. The control blocks created from the output of DBDGEN and PSBGEN, e.g., a DMB of an internal PSB created by the ACB utility program.

application control block generation (ACBGEN). The process by which application control blocks are created.

application control table (ACT). A DL/I online table describing those CICS application programs that use DL/I and the PSBs they are authorized to use.

argument. (1) (ISO)[10] An independent variable. (2) (ISO)[10] Any value of an independent variable. (3) Information, such as names, constants, or variable values included within the parentheses in a DL/I command.

10 International Organization for Standardization, Technical Committee 97/Subcommittee 1

attribute. A property of an entity expressing a value. Synonymous with *field.*

B

backout. The process of removing all the data base updates performed by an application program that has terminated abnormally. See also *dynamic backout.*

batch checkpoint/restart. The facility that enables batch processing programs to synchronize checkpoints and to be restarted at a user-specified checkpoint.

batch processing. A processing environment in which data base transactions requested by applications are accumulated and then processed periodically against a data base.

Boolean operator. (1) (ISO)[10] An operator, each of the operands of which and the result of which takes one of two values. (2) An operator that represents symbolically relationships, such as AND, OR, and NOT, between entities.

business process. A defined function of a business enterprise usually interrelated through information requirements with other business processes. For example, personnel management is the business process responsible for employee welfare from pre-hire through retirement. It is related to the accounting business process through payroll.

C

CA. Control area.

call. (1) (ISO)[10] The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) (ISO)[10] In computer programming, to execute a call. (3) The instruction in the COBOL, PL/I, or Assembler program that requests DL/I services. For RPG II, see *RQDLI command.* See also *command.*

checkpoint. A time at which significant system information is written on the system log, and optionally, the system is shut down.

child. Synonymous with *child segment.*

child segment. A segment one level below the segment which is its parent, with a direct path back up to the parent. Depending on the structure of the data base, a parent may have many children; however, a child has only one parent segment. Referring to Figure A-1:

- all the B, D, and F segments are children of A001.
- C5 and C7 are children of B01, but not children of the other B segments.
- B02 has no children.

See also *logical child* and *physical child.*

CI. Control interval.

combined checkpoint. In MPS batch programs, a VSE checkpoint followed immediately by a DL/I CHKP call. The two checkpoints together form one logical checkpoint and allow the use of the VSE Restart facility to restart MPS batch programs.

command. The statement in DL/I High Level Programming Interface (HLPI) that requests services for application programs written in COBOL or PL/I. See also *call.*

command code. An optional addition to the SSA that provides specification of a function variation applicable to the call function.

concatenated key. The key constructed to access a particular segment. It consists of the key fields, including that of the root segment and successive children down to the accessed segment.

control area (CA). A collection of control intervals. Used by VSAM to distribute free space.

control interval (CI). (1) A fixed length amount of auxiliary storage space in which VSAM stores records and distributes free space. (2) The unit of information transmitted to or from auxiliary storage by VSAM.

D

data aggregate. A group of data elements that describe a particular entity. Synonymous with *segment.* See also *data element.*

data base (DB). (1) (ISO)[11] A set of data, part of the whole of another set of data, and consisting of at least one file, that is sufficient for a given purpose or for a given data processing system. (2) A collection of data records comprised of one or more data sets. (3) A collection of interrelated or independent data items stored together without unnecessary redundancy to serve one or more applications. See *physical data base* and *logical data base.*

data base administration (DBA). The tasks associated with defining the rules by which data is accessed and stored. The typical tasks of data base administration are outlined in the *DL/I DOS/VS Data Base Administration*, SH24-5011.

data base administrator (DBA). A person in an installation who has the responsibility (full or part time) for technically supporting the use of DL/I.

data base description (DBD). A description of the physical characteristics of a DL/I data base. One DBD is generated and cataloged in a core image library for each data base that is used in the installation. It defines the structure, segment keys, physical organization, names, access method, devices, etc., of the data base.

data base integrity. The protection of data items in a data base while they are available to any application program. This includes the isolation of the effects of concurrent updates to a data base by two or more application programs.

data base organization. The physical arrangement of related data on a storage device. DL/I data base organizations are hierarchical direct (HD) and hierarchical sequential (HS). See *hierarchical direct organization* and *hierarchical sequential organization.*

---

11 International Organization for Standardization, Technical Committee 97/Subcommittee 1

data base record. A collection of DL/I data elements called segments hierarchically related to a single root segment.

Referring to Figure A-1: A001, B01,C5, C7, B02, B03, C2, D, E, F,F,F constitute a data base record.

data base reorganization. The process of unloading and reloading a data base to optimize physical segment adjacency, or to modify the DBD.

data communication (DC). A program that provides terminal communications and automatic scheduling of application programs based on terminal input. For example, CICS/DOS/VS.

data dictionary. (1) A centralized repository of information about data, such as its meaning, relationship to other data, usage, and format. (2) A program to assist in effectively planning, controlling, and evaluating the collection, storage, and use of data. For example, DOS/VS DB/DC Data Dictionary.

data element. The smallest unit of data that can be referred to. Synonymous with *field.* See also *data aggregate.*

data field. Synonymous with *field.*

data independence. (1) The concept of separating the definitions of logical and physical data such that application programs do not depend on where or how physical units of data are stored. (2) The reduction of application program modification in data storage structure and access strategy.

data management block (DMB). The data management block is created from a DBD by the application control blocks creation and maintenance utility, link edited, and cataloged in a core image library. The DMB describes all physical characteristics of a data base. Before an application program using DL/I facilities can be run, one DMB for each data base accessed, plus a PSB for the program itself, must be cataloged in a core image library. The DMBs and the associated PSB are automatically loaded into main storage from the core image library at the beginning of the application program execution (their loading is controlled by the parameter information supplied to DL/I at the beginning of program execution).

data set. A named organized collection of logically related records. They may be organized sequentially, as in the case of DOS/VSE SAM, or in key entry sequence, as in the case of VSE/VSAM. Synonymous with *file.*

data set group (DSG). A control block linking together a data base with the data sets comprising this DL/I data base.

DB. Data base.

DBA. (1) Data base administration. (2) Data base administrator.

DBD. Data base description.

DBDGEN. Data base description generation -- the process by which a DBD is created.

DB/DC. Data base/data communication.

DC. Data communication.

dependent segment. A DL/I segment that relies on at least the root segment (or on another segment at a level immediately above its own) for its full hierarchical meaning. Synonymous with *child segment.*

destination parent. The physical or logical parent segment reached by the logical child path.

device independence. The concept of writing application programs such that they do not depend on the physical characteristics of the device on which data is stored.

DIB. DL/I interface block.

direct access. The retrieval or storage of a VSAM data record independent of the record's location relative to the previously retrieved or stored record. (See also *address direct access* and *keyed direct access*). Contrast with *sequential access.*

distributed data. The ability of DL/I application programs to access a data base that is resident on another processor.

distributed free space. See *free space.*

DL/I documentation aid. An extension to ACBGEN that collects DBD and PSB information and stores it in SQL/DS tables. This information can subsequently be accessed directly by ISQL.

DL/I interface block (DIB). Variables automatically defined in an application program using HLPI to receive information passed to the program by DL/I during execution. Contrast with *PCB mask.*

DMB. Data management block.

DSG. Data set group.

DTF. Define the file -- a control block that connects a program to a SAM data set.

dynamic backout. A process that automatically cancels all activities performed by an application program that terminates abnormally.

E

entity. An item about which information is stored. It has properties that can be recorded. Information about an entity is a record.

entry sequenced data set (ESDS). A VSAM data set whose records are physically in the same order as they were put in the data set. It is processed by addressed direct access or addressed sequential access and has no index. New records are added at the end of the data set.

ESDS. Entry sequenced data set

exclusive intent. The scheduling intent type that prevents an application program from being scheduled concurrently with another application program. See *scheduling intent.*

F

FDB. field description block

field. (1) (ISO)[10] In a record, a specified area used for a particular category of data, for example, in which a salary rate is recorded. (2) a unique or nonunique area (as defined during DBDGEN) within a segment that is the smallest unit of data that can be referred to. (3) Any designated portion of a segment. (4) See also *key field.*

field level sensitivity. The ability of an application program to access data at the field level. See *sensitivity.*

file. (ISO)[10] A set of related records treated as a unit. See also *data set.*

forward. Movement in a direction from the beginning of the data base to the end of the data base, accessing each record in ascending root key sequence, and accessing the dependent segments of each root segment from top to bottom and from left to right. Referring to Figure A-1 , forward accessing of all segments shown would be in the following sequence:

A001, B01, C5, C7, B02, B03, C2, D, E, F, F, F, A002.

free space. Space available in a VSAM data set for inserting new records. The space is distributed throughout a key sequenced data set (KSDS) or left at the end of an entry sequenced data set (ESDS). Synonymous with *distributed free space.*

free space anchor point. A fullword at the beginning of a control interval pointing to the first free space element in this CI.

free space element. In HD data bases, the portions of direct access storage not occupied by DL/I segments are called and marked as free space elements.

FSA. free space anchor point

FSE. free space element

H

HD. Hierarchical direct.

HDAM. Hierarchical direct access method.

HIDAM. Hierarchical indexed direct access method.

HIDAM index. A data base that consists of logical DL/I records each containing an image of the key field of a HIDAM root segment. A HIDAM index data base consists of one VSAM KSDS (keyed sequenced data set).

hierarchical sequence. The sequence of segment occurrences in a data base record defined by traversing the hierarchy from top to bottom, left to right, and front to back.

hierarchical direct access method (HDAM). Provides for direct access to a DL/I data base in the HD organization. Segments are stored in VSAM control intervals and are referenced by a relative byte address. Root segments are accessed through a randomizing routine. An HDAM data base consists of one VSAM entry sequenced data set (ESDS).

hierarchical direct organization. An organization of DL/I segments of a data base that are related by direct addresses and may be accessed through an HD randomizing routine or an index.

hierarchical indexed direct access method (HIDAM). Provides for indexed access to a DL/I data base in the HD organization. Segments are stored in VSAM control intervals and are referenced by a relative byte address. Root segments are accessed through a HIDAM index data base. A HIDAM data base consists of one VSAM Entry Sequenced Data Set (ESDS) and its associated index.

hierarchical indexed sequential access method (HISAM). Provides for indexed access to a DL/I data base. A HISAM data base consists of one VSAM key sequenced data set (KSDS) and one VSAM entry sequenced data set (ESDS).

hierarchical sequential access method (HSAM). The segments of a DL/I HSAM physical data base record are arranged in sequential order with the root segments followed by the dependent segments. HSAM data bases are accessed by the DOS/VSE sequential access method (SAM).

hierarchical sequential organization. An organization of DL/I segments of a data base that are related by physical adjacency.

hierarchy. (1) An arrangement of data segments beginning with the root segment and proceeding downward to dependent segments. (2) A "tree" structure.

high level programming interface (HLPI). A DL/I facility providing services to application programs written in either COBOL or PL/I Optimizer language through commands.

HISAM. Hierarchical indexed sequential access method.

HLPI. High level programming interface.

HS. Hierarchical sequential.

HSAM. Hierarchical sequential access method.

I

index data base. An ordered collection of DL/I index entries (segments) consisting of a key and a pointer used by VSAM to sequence and locate the records of a key sequenced data set (KSDS). Organized as a balanced tree of levels of index.

index data set. Synonymous with *index data base.*

index pointer segment. The segment that contains the data and pointers used to index the index target segments.

index record. A system-created collection of VSAM index entries that are in collating sequence by the key in each of the entries.

index segment. The segment in the index data base that contains a pointer to the segment containing data (the indexed segment). Synonymous with *index pointer segment.*

index set. The set of VSAM index levels above the sequence set. An entry in a record in one of these levels contains the highest key entered in an index record in the next lower level and a pointer that indicates the record's physical location.

index source segment. The segment containing the data from which the indexing segment is built.

index target segment. The segment pointed to by a secondary index entry, that is, by an index pointer segment.

indexed segment. A segment that is located by an index. Synonymous with *index target segment.*

intersection data. Any user data in a logical child segment that does not include the logical parent's concatenated key.

inverted file. In information retrieval, a method of organizing a cross-index file in which a key identifies a record. The items pertinent to that key are indicated.

ISPF. Interactive System Productivity Facility. A dialog manager for interactive applications. It provides control and services to support execution of the dialogs.

ISQL. The Interactive Structured Query Language is part of SQL/DS. It enables terminal users to work directly with the data without writing a program.

ISQL extract defines utility. This is a utility that creates ISQL EXTRACT DEFINE commands automatically to be used to define a DL/I data base to the extract facility of SQL/DS.

K

key. (1) (ISO)[10] One or more characters within a set of data that contains information about that set, including its identification. (2) The field in a segment used to store segment occurrences in sequential order. (3) A field used to search for a segment. See *primary key* and *secondary key.* (4) Synonymous with *key field* and *sequence field.*

Note: A segment may or may not have a key, that is, a sequence field. All root segments, except for HSAM and simple HSAM data bases, must have keys. DL/I ensures that multiple segments of the same type that have keys are maintained in strict ascending sequence by key. The key may be located anywhere within a segment; it must be in the same location in all segments of the same type within a data base. The maximum sizes for keys are 236 alphameric characters for root segments, and 255 for all dependent segments. Keys provide a convenient way to retrieve a specific occurrence of a segment type, maintain the uniqueness and sequential integrity of multiples of the same segment type, and determine under which segment of a group of multiples new dependent segments are to be inserted. Keys should normally be prescribed for all segment types; the exception being if there will never be multiples of a particular type or if a particular segment type will never have dependents.

key field. The field in a segment used to store segment occurrences in sequential ascending order. A key field is also a search field. Synonymous with *key* and *sequence field.*

key sequenced data set (KSDS). A VSAM file whose records are loaded in key sequence and controlled by an index. See also *keyed direct access* and *keyed sequential access.*

keyed direct access. The retrieval or storage of a data record by use of an index that relates the record's key to its physical location in the VSAM data set, independent of the record's location relative to the previously retrieved or stored record. See also *addressed direct access*, *keyed sequential access*, and *addressed sequential access.*

keyed sequential access. The retrieval or storage of a VSAM data record in its collating sequence relative to the previously retrieved or stored record, by the use of an index that specifies the collating sequence of the records by key. See also *addressed sequential access*, *keyed direct access*, and *keyed sequential access.*

KSDS. Key sequenced data set.

L

level. (1) (ISO)[10] The degree of subordination of an item in a hierarchic arrangement.
(2) Level is the depth in the hierarchical structure at which a segment is located.
Roots are always the highest level and the segments at the bottom of the structure are the lowest level.
The maximum number of levels in a a DL/I data base is 15. For purposes of documentation and reference, the levels are numbered from 1 to 15, with the root segments being level number 1. Referring to Figure A-1:

- Three levels are shown.
- The A segments (roots) are at the highest level (level 1).
- The C and E segments are at the lowest level (level 3).

local system. (1) A specific system in a multisystem environment. Contrast with *remote system.* (2) The system in a multisystem environment on which the application program is executing. The local application may process data from data bases located on both the same (local) system and and another (remote) system.

local view. A description of the data that an individual business process requires. See *system view.*

logical. When used in reference to DL/I components, logical means that the component is treated according to the rules of DL/I rather than physically as it may exist, or as it may be organized, on a physical storage device. For example, a logical DL/I record (a root segment and all of its dependent segments grouped) might be contained on several physical records or blocks on a storage device; and because of prior insertions and deletions, the segments might be in different physical sequence than that by which they are retrieved logically for the application program by DL/I.

logical child. A pointer segment that establishes an access path between its physical parent and its logical parent. It is a physical child of its physical parent; it is a logical child of its logical parent. See also *logical parent* and *logical relationship.*

logical data base. A data base composed of one or more physical data bases representing a hierarchical structure derived from relationships between data segments that can be different from the physical structure.

logical data base record. (1) A set of hierarchically related segments of one or more segment types. As viewed by the application program, the logical data base record is always a hierarchic tree structure of segments. (2) All of the segments that exist hierarchically dependent on a given root segment, and that root segment.

logical data structure. A hierarchic structure of segments that is not based solely on the physical relationship of the segments. See also *logical relationships.*

logical parent. The segment a logical child points to. A logical parent segment can also be a physical parent. See also *logical child* and *logical relationship.*

logical relationship. A user defined path between two segments; that is, between logical parent and logical child, which is independent of

any physical path. Logical relationships can be defined between segments in the same physical data base hierarchy or in different hierarchies.

logical twins. All occurrences of one type of logical child with a common logical parent. Contrast with *physical twin*. See also *twin segment*.

M

MPS. Multiple partition support.

MPS restart facility. The capability to restart an MPS batch job when a system or application program failure occurs, using VSE checkpoint/restart with the DL/I checkpoint command.

multiple partition support (MPS). Multiple partition support provides a centralized data base facility to permit multiple applications in different partitions to access DL/I data bases concurrently. MPS follows normal DL/I online conventions in that two programs cannot both update the same segment type in a data base concurrently. (With program isolation, two programs can concurrently update the same segment type; however, they cannot concurrently update the same segment. See *program isolation*.) However, two or more programs can retrieve from a data base while another program updates it. If one program has exclusive use of a data base, no other program can update it or retrieve from it.

multiple SSA. A series of segment search arguments (SSAs) included in a DL/I call to identify a specific segment or path. See also *segment search argument*.

O

object segment. The segment at the lowest hierarchical level specified in a particular command. See also *path call*.

online. A operating environment in which DL/I is used with CICS/DOS/VS (or another data communication program) to permit end-users of application programs to access and store information in a data base through terminals.

option. A command keyword used to qualify the requested function.

P

parent. Synonymous with *parent segment*.

parent segment. (1) A segment that has one or more dependent segments. Contrast with *child*. (2) A parent is the opposite of a child, or dependent segment, in that dependent segments exist directly beneath it at lower levels. A parent may also itself be a child. Referring to Figure A-1:

- A001 is the parent of all B, D, and F segments.
- D is a parent of E; yet a child of A.
- B02 is not a parent.
- None of the Level 3 segments are parents.

parentage. Establishment in a program of a particular parent as the beginning point for the use of the GNP (get next in parent) or GHNP (get hold next in parent) functions. Parentage can only be established by issuing successful GU, GHU, GN, or GHN calls; or GET UNIQUE or GET NEXT commands.

path. The chain of segments within a record that leads to the currently retrieved segment. The formal path contains only one segment occurrence from each level from the root down to the segment for which the path exists. The exact path for each retrieved segment is returned in the following fields of the PCB:

Field 2	Segment hierarchy level indicator
Field 6	Segment name feedback area
Field 7	Length of key feedback area
Field 9	Key feedback area, containing the concatenated keys in the path

Referring to Figure A-1:

- The path to C5 is A001, B01.
- The path to C7 is the same as the path to C5.
- There is no path to A002 because it is a root segment.

path call. (1) The retrieval or insertion of multiple segments in a hierarchical path in a single call, by using the D command code and multiple SSAs. (2) The retrieval, replacement, or insertion of data for multiple segments in a hierarchical path in a single command, by using the FROM or INTO options specifying an I/O area for each parent segment desired. The object segment is always retrieved, replaced, or inserted.

PCB. Program communication block.

PCB mask. A skeleton data base PCB in the application program by which the program views a hierarchical structure and into which DL/I returns the results of the application's calls.

physical child. A segment type that is dependent on a segment type defined at the next higher level in the data base hierarchy. All segment types, except the root segment, are physical children because each is dependent on at least the root segment. See also *child segment*.

physical data base. An ordered set of physical data base records.

physical data base record. A physical set of hierarchically related segments of one or more segment types.

physical data structure. A hierarchy representing the arrangement of segment types in a physical data base.

physical parent. A segment that has a dependent segment type at the next lower level in the physical data base hierarchy. See also *parent*.

physical segment. The smallest unit of accessible data.

physical twins. All occurrences of a single physical child segment type that have the same (single occurrence) physical parent segment type. Contrast with *logical twins*. See also *twin segment*..

PI. Program isolation.

pointer. A physical or symbolic identifier of a unique target.

position pointer. For most call functions a position pointer exists before, during, and after the completion of the function. The pointer indicates the next segment in the data base that can be retrieved sequentially. It is normally set by the successful completion of the call function.

Referring to Figure A-1:

- If A001 has just been retrieved, the position pointer points to B01.
- If a new segment C6 has just been inserted, the position pointer points to C7.
- If the D segment has just been deleted (E will be deleted along with it), the position pointer points to the first F segment.
- If the last F segment has just been retrieved, it points to A002.

During PSB generation it is possible to specify either single or multiple positioning.

primary key. The data element, or combination of data elements, within a segment that uniquely identifies an occurrence of that segment. See *key* and *secondary key.*

program communication block (PCB). Every data base accessed in an application program has a PCB associated with it. The PCB actually exists in DL/I and its fields are accessed by the application program by defining their names within the application program as follows:

- COBOL - The PCB names are defined in the linkage section.
- PL/I - The PCB names are defined under a pointer variable.
- RPGII - The PCB names are automatically generated by the translator, or may be defined by the user.
- Assembler - The PCB names are defined in a DSECT.

There are nine fields in a PCB:

1. Data base name
2. Segment hierarchy level indicator
3. DL/I results status code
4. DL/I processing options
5. Reserved for DL/I
6. Segment name feedback area
7. Length of key feedback area
8. Number of sensitive segments
9. Key feedback area

program isolation (PI). A facility that isolates all data base activity of an application program from all other application programs active in the system until that application program commits, by reaching a synchronization point, that the data it has modified or created is valid.

This concept makes it possible to dynamically backout the data base activities of an application program that terminates abnormally without affecting the integrity of the data bases controlled by DL/I. It does not affect the activity performed by other application programs processing concurrently in the system.

program specification block (PSB). A PSB is generated for each application program that uses DL/I facilities. The PSB is associated with the application program for which it was generated and contains a PCB for each data base that is to be accessed by the program. Once it is generated, the PSB is cataloged in a core image library, and subsequently processed by a utility along with the associated DBDs to produce the updated PSB and DMBs; all of these are cataloged in a core image library for subsequent use by the application program during execution.

PSB. Program specification block.

PSBGEN. PSB generation -- the process by which a program specification block is created.

Q

qualified call. A DL/I call that contains at least one segment search argument (SSA). See also *segment search argument.*

qualified segment selection. The identification of a specific occurrence of a given segment type in a command, by using the WHERE option in the command for the desired segment. Contrast with *qualified SSA.*

qualified SSA. A qualified segment search argument contains both a segment name that identifies the specific segment type, and segment qualification that identifies the unique segment within the type for which the call function is to be performed. See also *segment search argument* and *multiple SSA.*

R

RAP. Root anchor point.

RBA. Relative byte address.

read-only intent. The scheduling intent type that allows a program to be scheduled with any number of other programs except those with exclusive intent. No updating occurs. See *scheduling intent.*

record. A data base record is made up of at least a unique root segment, and all of its dependent segments. See *data base record.*

relative byte address (RBA). The displacement of a stored record or control interval from the beginning of the storage space allocated to the VSAM data set to which it belongs.

remote system. In a multisystem environment, the system containing the data base that is being used by an application program resident on another (local) system. Contrast with *local system.*

root anchor point (RAP). A DL/I pointer in an HDAM control interval that points to a root segment or a chain of root segments.

root segment. The highest level (level 1) segment in a record. A root segment must have a key unless the organization is HSAM or simple HSAM. The sequence of the root segments constitutes the fundamental sequence of the data base. There can be only one root segment per record. Dependent segments cannot exist without a parent root segment; but a root segment can exist without any dependent segments.

RQDLI command. The instruction in the RPG II program used to request DL/I services.

S

scheduling intent. An application program attribute defined in the PSB that specifies how the program should be scheduled if multiple programs are contending for scheduling. See *exclusive intent, read-only intent*, and *update intent*.

search field. In a given DL/I call, a field that is referred to by one or more segment search arguments (SSAs).

secondary index. Secondary indexes can be used to establish alternate entries to physical or logical data bases for application programs. They can also be processed as data bases themselves. See also *secondary index data base*.

secondary index data base. An index used to establish accessibility to a physical or logical data base by a path different from the one provided by the data base definition. It contains index pointer segments.

secondary key. A data element, or combination of data elements, within a segment that identifies -- and is used to locate -- those occurrences of the segment that have a property named by the key. See *key* and *primary key*.

segment. A segment is a group of similar or related data that can be accessed by the application program with one I/O function call. There may be a number of segments of the same type within a record.

segment name. A segment name is assigned to each segment type. Segment names for the different segment types must be unique within a data base. The segment name is used by the application programmer when constructing a qualified or unqualified SSA prior to issuing a call for a specific segment. Synonymous with *segment type*.

segment occurrence. One instance of a set of similar segments.

segment search argument (SSA). Describes the segment type, or specific segment within a segment type, that is to be operated on by a DL/I call. See also *multiple SSA*, *qualified SSA*, and *unqualified SSA*.

segment selection. The specifying of parent and object segments by name in a command. Selection may be either qualified or unqualified. Contrast with *segment search argument*.

segment type. A user-defined category of data. Referring to Figure A-1, there are six different types of segments: A through F.

Different segment types may have different lengths, but within each single type, all segments must be the same length (unless variable length segments have been specified by DBA). Synonymous with *segment name*.

sensitivity. (1) A DL/I capability that ensures that only data segments or fields predefined as "sensitive" are available for use by a particular application program. The sensitivity concept also provides a degree of control over data security, inasmuch as users can be prevented from accessing particular segments or fields from a logical data base. (2) Sensitivity to the various segments and fields that constitute a data base is controlled, on a program-by-program basis, when the PSB for each program is generated. For example, a program is said to be sensitive to a segment type when it can access that segment type. When a program is not sensitive to a particular segment type, it appears to the program as if that segment type does not exist at all in the data base. Segment sensitivity applies to types of segments, not to specific segments within a type, and to all segment types in the path to the lowest level sensitive segment type.

sequence field. Synonymous with *key field*.

sequence set. The lowest level of a VSAM index. It immediately controls the order of records in a key sequenced data set (KSDS). A sequence set entry contains the key of the highest keyed record stored in a control interval of the data set and a pointer to the control interval's physical location. A sequence set record also contains a pointer to the physical location of each free control interval in the fan-out of the record.

sequential processing. Processing or searching through the segments in a data base in a forward direction. See also *forward*.

simple HISAM. A hierarchical indexed sequential access method data base containing only one segment type.

source segment. A segment containing the data used to construct the secondary index pointer segment. See also *secondary index data base*.

SQL/DS. The Structured Query Languate/Data System is a relational data base management system designed for end users. SQL/DS enables an end user to access data in online, interactive, and batch systems.

SSA. segment search argument.

status code. Each DL/I request for service returns a status code that reflects the exact result of the operation. The first operation that a program should perform immediately following a DL/I request is to test the status code to ensure that the function requested was successful. Following a command, the status code is returned in the DIB at the label DIBSTAT. Following a call, the status code is returned in field 3 of the PCB.

sync(h) point. Synonymous with *synchronization point*.

synchronization point. A logical point in time during the execution of a application program where the changes made to the data bases by the program are committed and will not be backed out. Synonymous with *sync point* or synch point.

A synchronization point is created by:

- a DL/I CHECKPOINT command or CHKP call
- a DL/I TERMINATE command or TERM call
- a CICS/VS synch point request
- an end of task (online) or an end of program (MPS-batch).

system view. A conceptual data structure that integrates the individual data structures associated with local views into an optimum arrangement for physical implementation as a data base. See *local view*.

T

transaction. A specific set of input data that triggers the execution of a specific process or job.

twin segments. All child segments of the same segment type that have a particular instance of the same parent type. See also *physical twins* and *logical twins*.

twins. Synonymous with *twin segments.*

U

unqualified call. A DL/I call that does not contain a segment search argument.

unqualified segment selection. The identification of a given segment type in a command without specifying a particular occurrence of that segment type (without using the WHERE option in an HLPI command). As a general rule, unqualified segment selection retrieves the first occurrence of the specified segment type. Contrast with *unqualified SSA.*

unqualified SSA. An unqualified SSA contains only a segment name that identifies the specific type of segment for which the I/O function is to be performed. As a general rule, the use of an unqualified SSA retrieves the first occurrence of the specified type of segment. See also *segment search argument.*

update intent. The scheduling intent type that permits application programs to be scheduled with any number of other programs except those with exclusive intent. If program isolation is used, update intent permits application programs to be scheduled only with any number of other programs with read-only intent. See *scheduling intent.*

UPSI. User program switch indicator. A special 8 bit byte that allows each bit to be programmed by the user as "1" or "0". Bits may be read by a program to determine what the user wants to do.

Definition of Manual Codes

ADBD *DL/I DOS/VS Application and Data Base Design,* SH24-5022

CALL *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces,* SH12-5411

DBA *DL/I DOS/VS Data Base Administration,* SH24-5011

DIAG *DL/I DOS/VS Diagnostic Guide,* SH24-5002

GIM *DL/I DOS/VS General Information,* GH20-1246

GNU *DL/I DOS/VS Guide for New Users,* SH24-5001

HLPI *DL/I DOS/VS Application Programming: High Level Programming Interface,* SH24-5009

IRDU *DL/I DOS/VS Interactive Resource Definition and Utilities,* SH24-5029

LLC *Low-Level Codes/Continuity Check in Data Language/I DOS/VS, Program Reference and Operations Manual,* SH20-9046

RDU *DL/I DOS/VS Resource Definition and Utilities,* SH24-5021

RRG *DL/I DOS/VS Recovery/Restart Guide,* SH24-5030

DL/I DOS/VS
Library Guide and Master Index
GH24-5008-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GH24-5008-1

Reader's Comment Form

Cut or Fold Along Line

DL/I DOS/VS Library Guide and Master Index (File No. S370/4300-50) GH24-5008-1

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

DL/I DOS/VS
Library Guide and Master Index
GH24-5008-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Cut or Fold Along Line

DL/I DOS/VS Library Guide and Master Index (File No. S370/4300-50) GH24-5008-1

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

DL/I DOS/VS
Library Guide and Master Index
GH24-5008-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GH24-5008-1

Reader's Comment Form

Cut or Fold Along Line

DL/I DOS/VS Library Guide and Master Index (File No. S370/4300-50) GH24-5008-1

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

GH24-5008-1

DL/I DOS/VS Library Guide and Master Index (File No. S370/4300-50) GH24-5008-1

GH24-5008-01