

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

Program Product

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

Program Number 5746-XX1

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font, with each letter formed by a series of horizontal bars of varying lengths, creating a striped effect.

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

Program Product

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

Program Number 5746-XX1

IBM

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

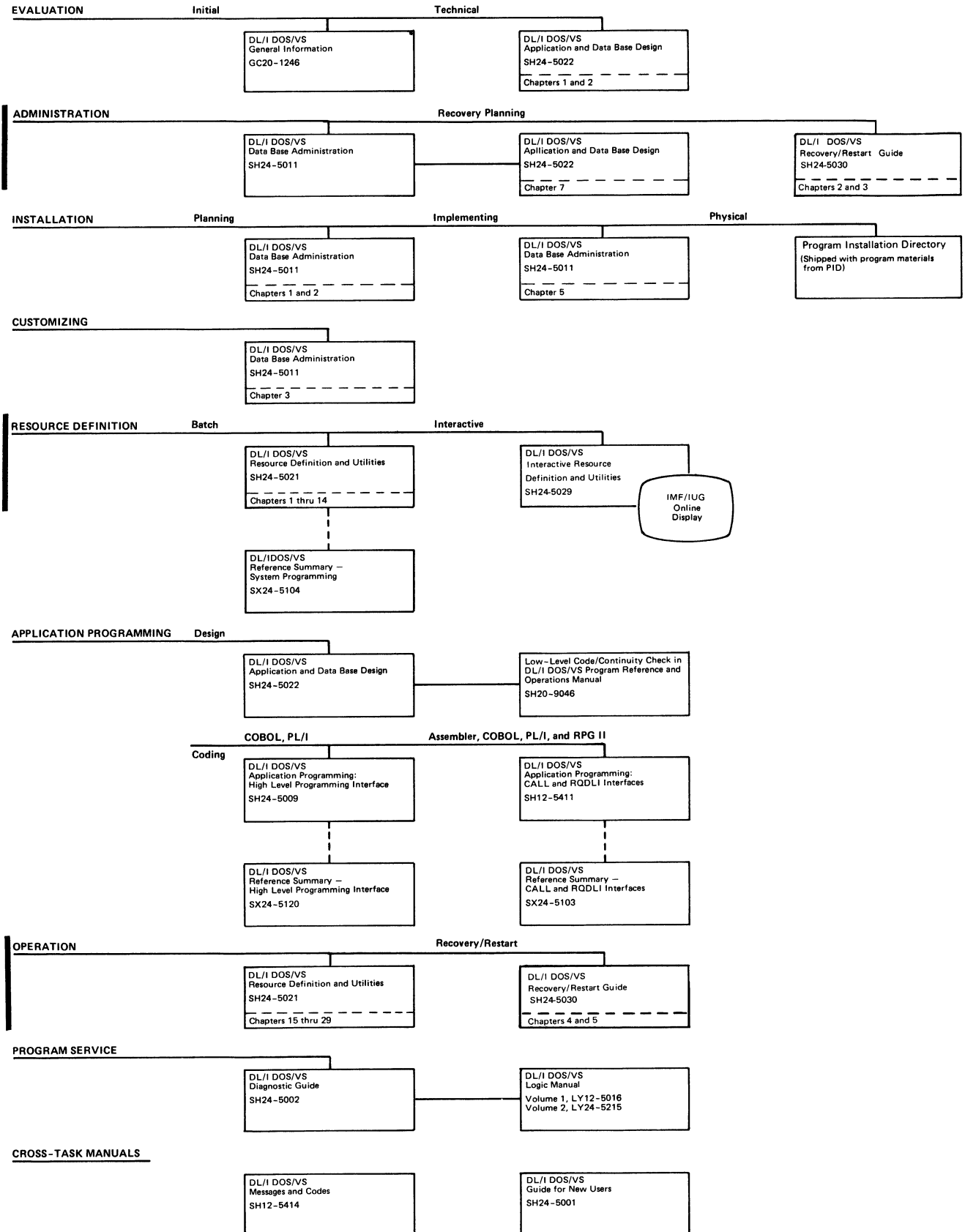


Figure 1. DL/I DOS/VS Library Organization

DL/I DOS/VS Publications

In order number sequence:

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

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 several 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	<p>Making fundamental decisions about procedures that are followed during the implementation tasks of installation, customization, operation, resource definition, application programming, and program service.</p> <p>Administration is an iterative task in that many of the decisions are made before installation, continually evaluated after installation, and revised as appropriate.</p>
Application Programming	<p>Designing, coding, compiling, executing, debugging, and testing application programs.</p>
Customization	<p>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.</p>
End Use	<p>Directly using an IBM program to achieve the purpose for which it is provided.¹</p>
Evaluation	<p>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.</p>
Installation	<p>Making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it, as required.</p>
Operation	<p>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.</p>
Program Service	<p>Describing, resolving, and correcting an IBM program problem. Program service encompasses collecting and examining problem related facts, organizing significant facts into a problem</p>

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	1-1
Evaluation	1-2
DL/I DOS/VS General Information, GH20-1246	1-2
Administration	1-3
DL/I DOS/VS Data Base Administration, SH24-5011	1-3
Installation	1-4
Customization	1-4
Resource Definition	1-5
DL/I DOS/VS Resource Definition and Utilities, SH24-5021	1-5
DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029	1-5
DL/I DOS/VS IMF and IUG Online Information	1-6
DL/I DOS/VS Reference Summary - System Programming, SX24-5104	1-7
Application Programming	1-8
DL/I DOS/VS Application and Data Base Design, SH24-5022	1-8
DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009	1-8
DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces, SH12-5411	1-9
Low-Level Code/Continuity Check in Data Language/I DOS/VS Program Reference and Operations Manual, SH20-9046	1-9
DL/I DOS/VS Reference Summary - High Level Programming Interface, SX24-5120	1-10
DL/I DOS/VS Reference Summary - CALL and RQDLI Interfaces, SX24-5103	1-10
Operation	1-11
DL/I DOS/VS Recovery/Restart Guide, SH24-5030	1-11
Program Service	1-12
DL/I DOS/VS Diagnostic Guide, SH24-5002	1-12
DL/I DOS/VS Logic, Volume 1, LY12-5016	1-13
DL/I DOS/VS Logic, Volume 2, LY24-5215	1-13
Cross-Task Manuals	1-14
DL/I DOS/VS Messages and Codes, SH12-5414	1-14
DL/I DOS/VS Guide for New Users, SH24-5001	1-15
Section 2. Master Index	2-1
Appendix A. Publication Cross-reference	A-1
How to Use This Appendix	A-1
DL/I DOS/VS General Information	A-3
DL/I DOS/VS System/Application Design Guide	A-5
DL/I DOS/VS Utilities and Guide for the System Programmer	A-8
Glossary	X-1
Definition of Manual Codes	X-11

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 recovering 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

/CK field types, defining **DBA**
/CK system-related field **RDU**
(see also **SUBSEQ** operand) **RDU**
/INSERT statement in **RPG II** **GNU**
/SX Field Types, defining **DBA**
/SX system-related field **RDU**
(see also **SUBSEQ** operand) **RDU**
"P", processing option **HLPI**

A

ABC creation and maintenance **GNU**
documentation aid **GNU**
abend **HLPI**
abend messages **DIAG**
abnormal ends in **IMF** **DIAG**
abnormal task termination, **DL/I** **DBA**
abnormal termination **GNU**
in batch **GNU**
in **CICS/VS** **GNU**
in **MPS** **GNU**
routine **GNU**
routines **GNU**
abnormal termination messages **HLPI**
abnormal termination routines
application of **DBA**
batch environment **DBA**
CICS/VS **DBA**
DL/I internal error termination **DBA**
MPS partition **DBA**
STXIT macros **DBA**
UPSI byte settings **DBA**
use of in batch **DBA**
use of in **MPS** batch **DBA**
ACB creation and maintenance for each **PSB** **GNU**
ACB creation and maintenance utility **DBA**
ACB extension, locating **DIAG**
ACB generation (**ACBGEN**) **DBA**
Documentation Aid **DBA**
Documentation Aid facility **DBA**
ACB, locating **DIAG**
ACBGEN **DIAG**
ACBGEN utility **RDU**
BUILD control statement **RDU**
description **RDU**
examples **RDU**
job control statements **RDU**
acceptable specification of arguments **HLPI**
access **DBA**
controlling **DBA**
encryption **DBA**
indexed **ADBD**
randomized **ADBD**
user field routine **DBA**
access by non-**DL/I** programs, controlling **DBA**
access method **GNU**
and physical data basis **GNU**
HD (hierarchical direct) **GNU**
HD characteristics **GNU**
hierarchical direct (**HD**) **GNU**
hierarchical direct access method (**HD**) **GNU**
hierarchical indexed sequential access
method (**HISAM**) **GNU**
HISAM (hierarchical indexed sequential

access method) **GNU**
HISAM considerations **GNU**
HISAM physical storage of a data base record **GNU**
HISAM (hierarchical sequential access method) **GNU**
HISAM considerations **GNU**
HISAM physical storage of a logical data structure **GNU**
overview **GNU**
simple hierarchical indexed sequential
access method (simple **HISAM**) **GNU**
simple hierarchical sequential access
method (simple **HISAM**) **GNU**
simple **HISAM** (simple hierarchical indexed sequential
access method) **GNU**
VSAM (virtual storage access method) **GNU**
access method control block, locating **DIAG**
access method services **GNU**
ALTER command **GNU**
DEFINE command **GNU**
use of **GNU**
DELETE command **GNU**
VERIFY command **GNU**
access method, choosing the **DBA**
access methods **DIAG**
characteristics **DBA**
direct organization **DIAG**
DL/I **DBA**
HD **DBA**
HS **DBA**
selection **DBA**
chart **DBA**
sequential organization **DIAG**
ACCESS operand **GNU**
DBD statement **GNU**
logical **DBD** statement **GNU**
ACCESS operand coding requirements **RDU**
HD data bases **RDU**
HDAM data bases **RDU**
HIDAM data bases **RDU**
HISAM data bases **RDU**
HISAM data bases **RDU**
LOGICAL data bases **RDU**
primary **INDEX** data bases **RDU**
secondary **INDEX** data bases **RDU**
SHISAM data bases **RDU**
SHSAM data bases **RDU**
access path **GNU**
data base **GNU**
in a data base record **GNU**
logical **GNU**
physical **GNU**
secondary indexes **GNU**
ACCESS statement **GNU**
format of **GNU**
HD indexed access **RDU**
HD randomized access **RDU**
HD secondary indexed access **RDU**
primary indexed access **GNU**
REF operand **GNU**
SEGM operand **GNU**
SEQFLD operand **GNU**
primary randomized access **GNU**
RMRTN operand **GNU**
SEGM operand **GNU**
SEQFLD operand **GNU**
SEQVAL operand **GNU**
secondary index **GNU**
access techniques **DBA**
for data base records **DBA**
direct processing **DBA**
sequential processing **DBA**

selection of **DBA**
 within data base records **DBA**
 direct processing **DBA**
 sequential processing **DBA**
 accessing data in SQL/DS tables **RDU**
 for Documentation Aid facility **RDU**
 acquiring data base space **DBA**
ACT (application control table) **GNU, RDU**
 (See also application control table)
 control statements formats **RDU**
 DLZACT TYPE=BUFFER **RDU**
 DLZACT TYPE=CONFIG **RDU**
 DLZACT TYPE=FINAL **RDU**
 DLZACT TYPE=INITIAL **RDU**
 DLZACT TYPE=PROGRAM **RDU**
 DLZACT TYPE=RPSB **RDU**
 control statements summary **RDU**
 data gathering **DIAG**
 default table **IRDU**
 variables **DIAG**
 definition **IRDU RDU**
 table variables **DIAG**
 example **RDU**
 functions **GNU**
 generation **DBA, DIAG, IRDU**
 errors **DIAG**
 job control statements **DIAG**
 how to define **RDU**
 input structure and rules **RDU**
 LIFO list **DIAG**
 linkage editor warning **DIAG**
 list table **IRDU**
 module flow **DIAG**
 online nucleus (DLZNUCxx) **GNU**
 specifying the end of **GNU**
 tables **DIAG**
ACT generation procedure
 CMS macro file **IRDU**
 definition **IRDU**
 generation **IRDU**
 VSE job stream **IRDU**
ACT macros
 relating to IMF panels **IRDU**
ACT worksheets **IRDU**
ACTGEN (application control table generation) **RDU, DIAG**
 definition **IRDU, RDU**
 description **IRDU**
 generation **IRDU, RDU**
 IMF **GNU**
 action modules **DBA**
 DL/I **DBA**
 real storage estimates **DBA**
 VSAM **DBA**
 adding new segment types to a data base structure **GNU**
 adding segments to an existing data base **GNU**
 additional PSB storage **DBA**
 administration and security, data **DBA**
 administration, data base user responsibility **GNU**
 advantages of LLC/CC in DL/I **LLC**
 AFTW routine trace point (ONLINEBH) **DIAG**
AI statement **RDU**
 aids
 batch initialization **DIAG**
 data location **DIAG**
 DL/I call **DIAG**
 logical retrieve debugging **DIAG**
 MPS batch initialization **DIAG**
 online remote data base **DIAG**
 online scheduling debugging **DIAG**
 online wait state debugging **DIAG**
 user error, batch and MPS **DIAG**
 AL field in an FSE **DIAG**
 ALIGN operand coding requirements **RDU**
 allocating data sets **DBA**
 ALT (application load table) **DBA, RDU**
 ALTER command, access method services **GNU**
 alternate processing sequence **CALL**
 alternate record entry point **DBA**
 AMXT parameter, CICS/VS **GNU**
 analysis, data base user responsibility **GNU**
 anchor point area **DIAG**
 for HD indexed or HIDAM data base **DIAG**
 for HD randomized or HDAM data base **DIAG**
AND, logical **CALL**
 expression of **CALL**
AO statement **RDU**
APAR requirements **DIAG**
 application analysis **ABDD**
 analyzing data **ABDD**
 creating a data dictionary **ABDD**
 defining programs **ABDD**
 example **ABDD**
 method **ABDD**
 requirements and considerations **ABDD**
 defining tasks **ABDD**
 naming conventions **ABDD**
 additional for DL/I **ABDD**
 in sample application **ABDD**
 introduction **ABDD**
 recommended **ABDD**
 application control block
 (See also ACBGEN)
 creation and maintenance **DIAG**
 creation errors **DIAG**
 maintenance errors **DIAG**
 application control blocks creation and maintenance
 (DLZUACB0) **GNU**
 BUILD statement **GNU**
 control statements **GNU**
 job control requirements **GNU**
 application control blocks creation and maintenance
 utility **RDU**
 application control blocks creation and maintenance
 utility - DLZUACB0 **DBA**
 application control table (ACT) **DBA, GNU, RDU**
 (See also ACT)
 generation (see ACTGEN) **DIAG, IRDU**
 specifying the end of **GNU**
 use **DIAG**
 scheduling call (online) **CALL**
 application data structure **GNU**
 designing **GNU**
 application design **ABDD, DBA**
 definition **ABDD**
 link to data base administration **ABDD**
 procedure **ABDD**
 application analysis **ABDD**
 combining local views **ABDD**
 converting to physical view **ABDD**
 creating local views **ABDD**
 implementing application **ABDD**
 obtaining application requirements **ABDD**
 preliminary analysis **ABDD**
 with data bases example **ABDD**
 application dumps **DIAG**
 application load table (ALT) **DBA, RDU**
 application program **CALL, GNU**
 communication with DL/I **CALL**
 data base processing functions **GNU**
 entry points **CALL**

- job control statements **CALL**
- sample **GNU**
- standard error routine **CALL**
- termination of **CALL**
- types of **GNU**
- application program (see user-written application program) **LLC**
- application program design **DBA**
- application program errors **DIAG**
 - batch and MPS batch **DIAG**
 - program check **DIAG**
- application programming with DL/I **GIM**
- application programming support **GIM**
 - low-level code and continuity check **GIM**
- application program online (see online application program) **CALL**
- application program parameter information **DIAG**
- application program relationship, describing **GNU**
- application program requirements **DBA**
 - for RPG II **GNU**
 - hints **DBA**
- application programs **DBA**
- application programs, Assembler language **LLC**
 - real storage estimates **DBA**
 - examples of call **LLC**
 - initial generation **LLC**
 - updating **LLC**
 - parameter list **LLC**
 - description of parameters **LLC**
 - initial generation **LLC**
 - updating **LLC**
 - requirements **LLC**
- :sample program **LLC**
- application programs, high-level languages **LLC**
 - COBOL **LLC**
 - examples of call, initial generation **LLC**
 - examples of call, updating **LLC**
 - sample program **LLC**
 - parameter list **LLC**
 - description of parameters **LLC**
- PL/I **LLC**
 - examples of call, initial generation **LLC**
 - examples of call, updating **LLC**
 - sample program **LLC**
- application requirements **ADBD**
 - definition **ADBD**
 - documentation **ADBD**
 - requirements listing **ADBD**
 - types of data **ADBD**
 - example **ADBD**
 - education company **ADBD**
 - sample program **ADBD**
 - existing sources **ADBD**
 - interviews **ADBD**
 - conducting **ADBD**
 - organizing **ADBD**
 - questionnaires **ADBD**
 - reviewing **ADBD**
 - setting up a team **ADBD**
- application view **DBA**
 - defining the **DBA**
 - definition **DBA**
 - of search fields **DBA**
 - physical data base **DBA**
- applications
 - installing additional **DBA**
 - data base considerations **DBA**
 - implementation considerations **DBA**
 - machine requirements **DBA**
 - storage requirements **DBA**

- applications under CICS/VS (PPT), defining **GNU**
- area, segment I/O **HLPI**
- argument **GNU, HLPI**
 - definition **HLPI**
 - examples **HLPI**
 - HLPI** use **GNU**
- argument list **CALL**
 - address of SSA(s) **CALL**
 - call function **CALL**
 - function and contents of **CALL**
 - I/O area address **CALL**
 - in DL/I call statement **CALL**
 - parameter count **CALL**
 - PCB address **CALL**
 - use of parameter count **CALL**
- ARSAabend debugging **DIAG**
- ASLOG **DIAG**
 - parameter statement **GNU, RDU, RRG**
- ASMTDLI **DIAG**
- Assembler application program **CALL**
 - combined checkpoint example **CALL**
 - examples of online calls **CALL**
- Assembler call **DIAG**
- Assembler language (see application programs, Assembler language) **LLC**
- Assembler language program **CALL**
 - batch program example **CALL**
 - DL/I call format **CALL**
 - entry points **CALL**
 - link-editing of **CALL**
 - online acquisition of working storage **CALL**
 - online call **CALL**
 - online scheduling call **CALL**
 - online termination call **CALL**
 - PCB mask **CALL**
 - termination of **CALL**
- Assembler listing, DBD generation **DIAG**
- Assembler macros, restriction of **CALL**
- assembly errors **DIAG**
- ASSGN parameters, entering **IRDU**
- ASSGN statement **HLPI**
 - HSAM data base **HLPI**
 - SHSAM data base **HLPI**
- asynchronous log **DBA, RRG**
 - advantage of **RRG**
 - option **DBA**
 - use with batch jobs **RRG**
- attributes, definition **GNU**
- audit procedures **LLC**
- auto report (RPG II) **GNU**
- AUTOLINK feature **CALL**
 - Assembler program **CALL**
 - COBOL program **CALL**
 - PL/I program **CALL**
 - RPG II program **CALL**
- automatic data format conversion **DBA, GNU**
- available DL/I functions, other **HLPI**
- available length field in an FSE **DIAG**

B

- backed up data, use for **RRG**
- backing out data changes **RRG**
- backing out incomplete LUWs **RRG**
- backing up a data base **RRG**
- backout **HLPI, RRG**
 - Data Base Backout utility **RRG**

- definition of **RRG**
- Dynamic Transaction Backout utility **RRG**
- Log Print utility **RRG**
- procedures for backout **RRG**
- requirements for **RRG**
- without forward recovery **RRG**
- backout and forward recovery **RRG**
 - for online recovery **RRG**
 - overhead **RRG**
 - strategy for **RRG**
- backout failure, batch procedures to recover **RRG**
- backout function, IUG recovery **DIAG**
- backout utility **DIAG, GNU, RRG**
 - data base **GNU**
 - use of **GNU**
- backup and restore **RRG**
- backup using Data Set Image Copy utility **RRG**
- backup/restore options **RRG**
 - data-oriented **RRG**
 - file-oriented **RRG**
 - volume-oriented **RRG**
- base data base, specifying **DBA**
- base segment for a concatenated segment, defining the **DBA**
- base segment for a simple segment, defining the **DBA**
- base segments, defining **DBA**
- basic **GNU**
 - PSB coding **GNU**
 - PSBs, sample **GNU**
- basic segment types in a hierarchical data structure **GNU**
- batch **GNU**
 - abnormal termination **GNU**
 - MPS programs, executing **GNU**
 - MPS, and online differences **GNU**
 - partition controller **GNU**
 - system, DL/I **GNU**
- batch and MPS batch application
 - dumps **DIAG**
 - program execution **CALL**
- batch application **GNU**
 - DL/I relationship **GNU**
 - program execution **GNU**
- batch application program **CALL**
 - job control example **CALL**
 - PL/I restrictions **CALL**
- batch application program, load **HLPI**
- batch checkpointing **RRG**
- batch communications area **DIAG**
- batch control flow **DIAG**
- batch data base processing **DBA**
- batch environment **CALL, DBA, CALL**
 - reinitializing during MPS restart **CALL**
- batch environment tracing (DLZTRACE) **DIAG**
 - sample DLZTRACE job **DIAG**
 - trace output **DIAG**
 - what information to trace **DIAG**
 - which calls to trace **DIAG**
- batch environment, comparing DL/I and VSE **CALL**
- batch initialization aids **DIAG**
- batch initialization parameters **IRDU**
- batch initialization/termination **DIAG**
- batch nucleus and program request handler, DL/I **DBA**
- batch partition controller **GNU**
- batch procedures for recovery from a backout failure **RRG**
- batch processing considerations **GNU**
- batch program **HLPI**
 - compilation and link-editing **HLPI**
 - entry **HLPI**
 - MPS **HLPI**
 - execution **HLPI**
 - job control statements **HLPI**
 - termination **HLPI**
 - MPS **HLPI**
- batch program execution **DBA**
- batch program structure **CALL, GNU**
 - Assembler **CALL, GNU**
 - COBOL **CALL, GNU**
 - PL/I **CALL, GNU**
 - RPG II **CALL, GNU**
- batch requirements **RDU**
 - job control statements **RDU**
 - parameter statement **RDU**
 - UPSI byte settings **RDU**
- batch system initialization **GNU**
- batch system storage, DL/I **DBA**
- batch termination **DIAG**
- batch trace output **DIAG**
- batch user error aids **DIAG**
- batch utility dumps **DIAG**
- batch wait states
 - BPC attach failure **DIAG**
 - scheduling conflict or MAXTASK **DIAG**
 - uncontrolled BPC abend **DIAG**
- BFPL (buffer pool control block prefix) **DIAG**
- BFPR (buffer prefix) **DIAG**
- BFRPOOL operand coding requirements **RDU**
- BFRPOOL parameter, DLZACT macro **GNU**
- BHINTF Type 1 (X'61') **DIAG**
- BHINTF Type 2 (X'62') **DIAG**
- bidirectional logical relationship (see LCHILD statement) **RDU**
- bidirectional logical relationships
 - defining a **DBA**
 - example of **DBA**
- bidirectional relationships **DBA**
- bidirectional versus unidirectional relationships **DBA**
- bill of material **LLC**
 - relationship **LLC**
- bill processor systems **LLC**
- binary halving method example **DBA**
- bit map **DIAG**
 - block **DIAG**
 - control interval **DIAG**
- block building **DIAG**
- BLOCK keyword, VSAM **LLC**
- BLOCK operand coding requirements **RDU**
 - HD data bases **RDU**
 - HDAM data bases **RDU**
 - HIDAM data bases **RDU**
 - HISAM data bases **RDU**
 - primary INDEX data bases **RDU**
 - secondary INDEX data bases **RDU**
 - SHISAM data bases **RDU**
- BLOCK operand, DATASET statement **GNU**
- block size, HD reorganization unload output **DIAG**
- Boolean **CALL**
 - logic qualification **CALL**
 - operator **CALL**
 - and SSA length **CALL**
- boolean expression support (see relational operator in qualified SSA) **CALL**
- Boolean operator restrictions with ISC **DIAG**
- Boolean operators **GNU, HLPI**
 - in WHERE option **GNU, HLPI**
- Boolean statement **CALL**
 - qualification **CALL**
- BPC attach failure **DIAG**
- braces, coding convention **HLPI**
- brackets, coding convention **HLPI**
- bridge (see Chained File - DL/I Bridge) **LLC**
- buf parameter **RDU**
- buffer allocation, VSAM **CALL**

buffer handler **DIAG**
 verifying input to **DIAG**
 buffer pool **CALL, DBA, CALL, GNU**
 activity statistics **DIAG**
 assignment **DBA**
 characteristics report **GNU**
 configuration **DBA**
 control block prefix (BFPL) **DIAG**
 control options **DBA**
 control options, specifying **GNU**
 finding data in **DIAG**
 locating the **DIAG**
 performance considerations **DBA**
 size (ACT) **GNU**
 statistics for batch **DBA**
 buffer prefix (BFPR) **DIAG**
 buffer statistic output **DBA**
 buffer statistics **DBA**
 buffer storage allocation **DBA**
 buffer subpools **DBA**
 buffer, locating **DIAG**
 BUILD control statement **DIAG, RDU**
 format **RDU**
 preparing **RDU**
 BUILD statement (ACBGEN) **GNU**
 building internal control blocks from DBDs and PSBs **RDU**
 building local views **ADBD**
 business process
 definition **ADBD**
 BYTES operand **GNU**
 FIELD statement **GNU**
 SEGM statement **GNU**
 SEGM statement for a logical child **GNU**
 VIRFLD statement (PSBGEN) **GNU**
 BYTES operand coding requirements **RDU**
 in field statement **RDU**
 /CK system-related fields **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 in SEGM statement **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 logical child segments **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 in SENFLD statement **RDU**
 in VIRFLD statement **RDU**

C

calculation specification **GNU**
 call **CALL**
 function **CALL**
 in online **CALL**
 parentage **CALL**
 position pointer **CALL**

call aids **DIAG**
 call execution, verifying **DIAG**
 call formats **DIAG**
 call functions **CALL**
 CHECKPOINT (see CHKP) **CALL**
 codes **DIAG**
 DELETE (see DLET) **CALL**
 GET HOLD NEXT (see GHN) **CALL**
 GET HOLD NEXT WITHIN PARENT (see GHNP) **CALL**
 GET HOLD UNIQUE (see GHU) **CALL**
 GET NEXT (see GN) **CALL**
 GET NEXT WITHIN PARENT (see GNP) **CALL**
 GET UNIQUE (see GU) **CALL**
 INSERT (see ISRT) **CALL**
 REPLACE (see REPL) **CALL**
 requiring SSAs **CALL**
 call paths
 delete/replace **DIAG**
 general **DIAG**
 insert **DIAG**
 insert (load type) request **DIAG**
 load **DIAG**
 replace **DIAG**
 retrieve **DIAG**
 call processing, online application **DIAG**
 call statement **CALL, LLC**
 format of **CALL, DIAG**
 warning on use of **LLC**
 call tracing (DLZTRACE) **DIAG**
 batch environment **DIAG**
 online environment **DIAG**
 CALLCON operand of DLZTRACE **DIAG**
 CALLCON parameters **DIAG**
 CALLDLI **DIAG**
 macro **CALL, DIAG, CALL**
 CALLFUNC **DIAG**
 CALLNUM **DIAG**
 calls
 (See also commands)
 Assembler **DIAG**
 COBOL **DIAG**
 data base **DIAG**
 formats **DIAG**
 functions **DIAG**
 delete - DLET **DIAG**
 get next (hold) - GN, GHN **DIAG**
 get next within parent (hold) - GNP, GHNP **DIAG**
 get unique (hold) - GU, GHU **DIAG**
 insert -ISRT **DIAG**
 replace - REPL **DIAG**
 online interface, CALLDLI **DIAG**
 online processing overview **DIAG**
 PL/I **DIAG**
 RPG II **DIAG**
 scheduling **DIAG**
 system **DIAG**
 terminating **DIAG**
 calls (RPG II) **GNU**
 to DL/I **GNU**
 with command codes **GNU**
 catalog entry for data base, checking **DIAG**
 catalog, VSAM **GNU**
 CBL statement **HLPI**
 CBLTDLI **DIAG**
 Chained File - DL/I Bridge **ADBD, LLC**
 compatibility lost if data stored in pointer segment **LLC**
 compatibility with LLC/CC in DL/I **LLC**
 change accumulation function, IUG recovery **DIAG**
 Change Accumulation utility **RRG**

data base GNU
 input to RRG
 output from RRG
 changing segment size DBA
 characteristics of DL/I data bases HLPI
 checking discontinued, reasons for LLC
 checking the response to a DL/I call in a CALL
 checklist for data base design GNU
 checklist, program planning HLPI
 checkpoint CALL, DBA, CALL, GNU, HLPI
 command GNU
 facility GNU
 facility, DL/I GNU
 format of GNU
 ID notification DBA
 ID verification DBA
 in batch MPS programs GNU
 in batch programs GNU
 in MPS batch programs GNU
 MPS with program isolation CALL
 notification CALL, GNU, HLPI
 during MPS restart GNU
 for MPS restart CALL, HLPI
 records in DL/I backout GNU
 synchronization GNU
 verification CALL, GNU, HLPI
 during MPS restart GNU
 for MPS restart CALL, HLPI
 CHECKPOINT call (see CHKP) CALL
 checkpoint command CALL, GNU, HLPI
 for MPS restart facility CALL, GNU, HLPI
 Assembler CALL
 COBOL CALL
 PL/I CALL
 format of RRG
 function HLPI
 in programming example HLPI
 using MPS restart facility HLPI
 parentage HLPI
 position pointer HLPI
 segment selection HLPI
 status codes HLPI
 syntax HLPI
 checkpoint facility DBA
 checkpoint records DBA
 illustrated DBA
 checkpoint request DBA
 in batch DBA
 in MPS batch DBA
 functions DBA
 use of DBA
 checkpoint function CALL
 definition of CALL
 checkpoint ID HLPI
 checkpoint request DBA
 checkpoint/restart, restrictions DBA
 checkpoint, restarting a program from a RRG
 checkpointing GNU, HLPI
 a data base GNU
 checkpoints, using RRG
 batch mode RRG
 CHECKPOINT command RRG
 examples of RRG
 MPS batch RRG
 required CICS/VS resources RRG
 saving backout time RRG
 VSE CHKPT macro RRG
 child GNU, HLPI
 coding logical GNU
 definition ADBD

logical GNU
 child and parent segments, connecting DBA
 child segment DBA
 definition DBA
 example DBA
 physical twins DBA
 twins DBA
 CHKP (checkpoint) GNU
 CHKP call CALL
 detailed description CALL
 I/O area usage CALL
 CHKPID option (RPG II) GNU
 CHKPT facility, DOS/VSE GNU
 choosing an access method ADBD
 direct access considerations ADBD
 logical relationships ADBD
 secondary indexes ADBD
 implementation requirements ADBD
 data access requirements ADBD
 performance considerations ADBD
 recovery requirements ADBD
 security requirements ADBD
 structural considerations ADBD
 choosing the DL/I log medium GNU
 CIANPT operand coding requirements RDU
 CICS HANDLE ABEND HLPI
 CICS/VS GNU
 abnormal termination GNU
 ACT (application control table), specifying the end of GNU
 AMXT parameter GNU
 and HLPI GNU
 CMXT parameter GNU
 controlling the number of tasks GNU
 CSTT transaction DBA
 DFHFCT (file control table) GNU
 DFHPCT (program control table) GNU
 DFHPPT (processing program table) GNU
 differences between batch, MPS, and online GNU
 DL/I table example GNU
 DL/I tables for the sample program GNU
 dynamic transaction backout GNU
 failure RRG
 emergency restart GNU
 EXEC translator GNU
 executing with MPS GNU
 extrapartition data set DIAG
 FCT (file control table) GNU
 file control table (FCT) DBA
 initialization DIAG
 integrity GNU
 internal error termination DBA
 intersystem communication support DBA, DIAG
 JCT (journal control table) GNU
 job control for creating the online nucleus GNU
 journal RRG
 journal control table (JCT) DBA
 mirror program (DFHMIR) DIAG
 MXT parameter GNU
 online environment, DL/I GNU
 online nucleus generation GNU
 assembly listing GNU
 control statement listing GNU
 diagnostics GNU
 output GNU
 PCT (program control table) GNU
 performance GNU
 PLT (program list table) GNU
 PPT (program processing table) GNU
 program check handler DIAG
 program control table (PCT) DBA

program list table (PLT) **DBA**
 program processing table (PPT) **DBA**
 programming considerations **GNU**
 releasing a PSB (RPG II) **GNU**
 requests **DBA**
 resources required for checkpointing **RRG**
 restart **GNU**
 restrictions **GNU**
 RQDLI commands **GNU**
 security **GNU**
 SIT (system initialization table) **GNU**
 sync-point record **GNU**
 system generation **GNU**
 system generation (DFHSG) **DBA**
 system initialization table (SIT) **DBA**
 system journal **DBA**
 system table preparation **GNU**
 table example **GNU**
 tables for the sample program, DLZSAM60 **GNU**
 task termination **DBA**
 tasks, controlling the number of **GNU**
 TCLASS parameter **GNU**
 temporary storage queue **DBA**
 TERMINATE command **GNU**
 termination **DIAG**
 termination functions **DBA**
 transactions, programming considerations **GNU**
 translator **GNU**
 TST (temporary storage table) **GNU**
 for MPS restart **GNU**
 UPSI byte settings **GNU**
 CICS/VS Command Interpreter **DIAG**
 CICS/VS command language environment **CALL**
 CICS/VS dynamic storage, use of **CALL**
 CICS/VS environment **CALL**
 data base call **CALL**
 scheduling call **CALL**
 termination call **CALL**
 CICS/VS EXEC translator **HLPI**
 CICS/VS Execution Diagnostic Facility (EDF) **HLPI**
 CICS/VS intersystem communication support **DBA**
 CICS/VS ISC to local DL/I interface **DIAG**
 CICS/VS ISC to remote DL/I PRH interface **DIAG**
 CICS/VS monitoring facilities (CMF) hooks **DBA**
 CICS/VS monitoring facilities, activating the **DBA**
 CICS/VS option of DLZTRACE OUTPUT operand **DIAG**
 CICS/VS synchronization point **HLPI**
 CICS/VS system generation requirements **DBA**
 file control table **DBA**
 journal control table **DBA**
 processing program table **DBA**
 program list table **DBA**
 system initialization table **DBA**
 CICS/VS system journal file **RRG**
 UPSI byte **RRG**
 CICS/VS trace table **CALL**
 CICS/VS transaction work area **CALL**
 use of **CALL**
 CICS/VS-DL/I tables, example of **RDU**
 ACT (application control table) **RDU**
 ALT (application load table) **RDU**
 DCT (destination control table) **RDU**
 FCT (file control table) **RDU**
 JCT (journal control table) **RDU**
 MCT (monitoring control table) **RDU**
 PCT (program control table) **RDU**
 PLT (program list table) **RDU**
 PPT (processing program table) **RDU**
 SIT (system initialization table) **RDU**
 SLC (storage layout control table) **RDU**
 TST (temporary storage table) **RDU**
 CICS/VS-DL/I tables, requirements **RDU**
 class data aggregate **ADBD**
 class schedule
 combining local view **ADBD**
 defining application programs **ADBD**
 distributed vs local processing **ADBD**
 frequency of use **ADBD**
 local view as input **ADBD**
 requirement for **ADBD**
 type of access **ADBD**
 clocks and counters **DBA**
 closing VSAM data sets **DBA, GNU**
 cluster **GNU**
 concept, VSAM **GNU**
 defining **GNU**
 CMAXTSK operand coding requirements **RDU**
 CMAXTSK parameter **GNU**
 DLZACT macro **GNU**
 CMS environment
 ACT macro file **IRDU**
 DBD macro file **IRDU**
 PSB macro file **IRDU**
 CMS/DOS environment, DL/I in **DBA**
 features of **DBA**
 restrictions of **DBA**
 CMXT call **DBA**
 example of **DBA**
 CMXT parameter, CICS/VS **GNU**
 COBOL **GNU**
 batch program structure **GNU**
 entry to an application program **GNU**
 online job control **GNU**
 online program structure **GNU**
 sample program names **GNU**
 SEGLENGTH option **GNU**
 COBOL (see application programs, high-level languages) **LLC**
 COBOL application program **CALL, GNU**
 batch program example **CALL**
 combined checkpoint example **GNU**
 combined checkpoints example **CALL**
 debug function **CALL**
 DL/I call format **CALL**
 entry points **CALL**
 error processing example **CALL**
 examples of online calls **CALL**
 job control for compile and link **CALL**
 link-editing of **CALL**
 online call **CALL**
 online scheduling call **CALL**
 online termination call **CALL**
 PCB mask **CALL**
 termination of **CALL**
 COBOL batch control flow **DIAG**
 COBOL batch program structure **CALL**
 COBOL call **DIAG**
 COBOL MPS batch control flow **DIAG**
 COBOL online control flow (CICS/VS) **DIAG**
 code assignment **LLC**
 consecutive during adding to data base **LLC**
 once after loading data base **LLC**
 codes, segment flag **DIAG**
 coding **GNU**
 a logical DBD **GNU**
 conventions, DBDGEN **GNU**
 logical relationship in a physical DBD **GNU**
 PSBs for logical data bases **GNU**
 PSBs for secondary indexes **GNU**

the object segment selection **GNU**
 the PCB option **GNU**
 coding conventions **CALL, HLPI, RDU**
 format 1 **RDU**
 format 2 **RDU**
 combined checkpoint **CALL, GNU, HLPI**
 COBOL example **GNU**
 example **CALL**
 Assembler **CALL**
 COBOL **CALL**
 PL/I **CALL**
 for MPS restart facility **CALL, GNU, HLPI**
 in programming example **HLPI**
 PL/I example **GNU**
 programming example **GNU**
 use in MPS restart **DBA**
 verification **DBA, HLPI**
 combined checkpoints **CALL**
 combining local views **ADBD**
 command codes **CALL, GNU**
 applicability table **CALL**
 D **CALL**
 F **CALL**
 format of SSAs with **CALL**
 in SSA **CALL, GNU**
 L **CALL**
 N **CALL**
 null (dummy) **CALL**
 Q **CALL**
 U **CALL**
 use of **CALL**
 V **CALL**
 command delimiters **GNU**
 command example **GNU**
 COBOL **GNU**
 HLPI **GNU**
 PL/I **GNU**
 Command Interpreter, CICS/VS **DIAG**
 command language **HLPI**
 command language environment **CALL**
 command statements **GNU**
 overview **GNU**
 command syntax **DIAG**
 command-delimiter **HLPI**
 definition **HLPI**
 description **HLPI**
 examples **HLPI**
 in syntax **HLPI**
 commands **GNU**
 (See also calls)
 and data base positioning **GNU**
 CHECKPOINT **GNU**
 contents of **GNU**
 DELETE **GNU**
 descriptions of **GNU**
 establishing position **GNU**
 examples **DIAG**
 GET NEXT **GNU**
 GET NEXT IN PARENT **GNU**
 GET UNIQUE **GNU**
 HLPI **DIAG**
 INSERT **GNU**
 issued for **GNU**
 kinds of **GNU**
 LOAD **GNU**
 REPLACE **GNU**
 syntax **DIAG**
 comment statement format **DIAG**
 communication region restrictions **CALL**
 communications area, batch **DIAG**

comparative value **GNU**
 SSA **GNU**
 comparative value in qualified SSA **CALL**
 compare statement format (PCB compare) **DIAG**
 compilation and link editing, online **GNU**
 compilation and link-editing **HLPI**
 COBOL **HLPI**
 PL/I **HLPI**
 compilation and translation, RPG II **GNU**
 compile and link edit, RPG II **GNU**
 compile and link-edit job control for application program **CALL**
 component item (see component parts) **LLC**
 component parts **LLC**
 compression routine (see COMPRTN operand) **RDU**
 compression, segment **DBA**
 COMPRTN operand coding requirements **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 COMPRTN operand, SEGM statement **GNU**
 COMREG restriction **GNU, HLPI**
 COMREG, restrictions **CALL**
 concatenated key **DBA, GNU, HLPI**
 accessing a logical child **GNU**
 data elements identified by whole **ADBD**
 definition **ADBD**
 destination parent (DPCK) **GNU**
 example of **RDU**
 in key feedback area **CALL**
 intersection data **ADBD**
 introducing to avoid many-to-many mapping **ADBD**
 key length, specifying the longest **DBA**
 logical parent (LPCK) **GNU**
 concatenated segment **DBA, GNU, HLPI**
 as variable length segment **HLPI**
 format **GNU**
 in LOAD command **HLPI**
 in logical data bases **HLPI**
 included in SEGLENGTH **HLPI**
 logical relationships **GNU**
 concatenated segments (see SOURCE operand coding requirements for LOGICAL data bases) **RDU**
 concept of data bases **GIM**
 advantages of data bases **GIM**
 concepts **GNU**
 data base **GIM, GNU**
 of data base design **GNU**
 of data elements **GNU**
 concepts and terminology **DBA**
 child segment **DBA**
 data base **DBA**
 data base record **DBA**
 dependent segment **DBA**
 fields **DBA**
 general terminology **DBA**
 hierarchy **DBA**
 level **DBA**
 parent segment **DBA**
 root segment **DBA**
 segment **DBA**
 segment occurrence **DBA**
 segment type **DBA**
 terminology **DBA**
 twin segment **DBA**
 concepts of recovery **RRG**
 conceptual data structure **ADBD**
 design of **ADBD**
 included in local view **ADBD**
 not a data base **ADBD**
 conceptual structure

data base as **ABDD**
 system view as **ABDD**
 concurrent DL/I tasks, execution of (ACT) **GNU**
 concurrent updates, program isolation **GNU**
 configuration, system **LLC**
 connecting parent and child segments **DBA**
 connection technique, defining the **DBA**
 considerations **GNU, HLPI**
 restart **GNU, HLPI**
 considerations and restrictions, general **HLPI**
 contention control, processing **DBA**
 contention management **DBA**
 program isolation **GNU**
 contents of a data base **GNU**
 continuation statements **RDU**
 continuation statements, parameter **DIAG**
 continuity checking **LLC**
 during initial generation **LLC**
 reasons for **LLC**
 uses of **LLC**
 violations detected by **LLC**
 control block relationships **DIAG**
 control blocks **DBA**
 DL/I **DBA**
 VSAM **DBA**
 control blocks in the DL/I nucleus **DIAG**
 control data base initialization **LLC**
 requirements **LLC**
 sample **LLC**
 control flow
 batch **DIAG**
 MPS batch **DIAG**
 tracing **DIAG**
 control interval
 and block sizes **DBA**
 VSAM **GNU, LLC**
 control procedures **LLC**
 control statement formats for DLZDLTXX **DIAG**
 control statement requirements **DIAG**
 trace print utility **DIAG**
 TI statement **DIAG**
 TO statement **DIAG**
 control statements **IRDU**
 control statements, basic DBDGEN **GNU**
 control table, IUG created **DIAG**
 CONTROL tables **IRDU**
 controlling data access **DBA**
 field level sensitivity **DBA**
 masking segments **DBA**
 segment sensitivity **DBA**
 controlling key **ABDD**
 controlling the number of CICS/VS and DL/I tasks **GNU**
 controlling the number of tasks **GNU**
 conventional file
 compared to data base **ABDD**
 implementation compared to data base **ABDD**
 physical record **ABDD**
 processing vs data base processing **ABDD**
 conventions for DL/I, additional naming **DBA**
 conventions, coding **HLPI**
 conversion, automatic data format **DBA**
 converting DL/I tables from IPF to ISPF format **IRDU**
 converting HLPI commands to calls **GNU**
 converting IMF tables to ISPF **IRDU**
 converting system view to physical view **ABDD**
 choice of access method **ABDD**
 data base organization and access methods **ABDD**
 DBA decisions **ABDD**
 dependent segment access **ABDD**
 root segment access **ABDD**

 updating data **ABDD**
 direct organization **ABDD**
 sequential organization **ABDD**
 copy disk volume **RRG**
 functional support **RRG**
 multivolume file dump/copy/restore **RRG**
 NOREWIND OPTION **RRG**
 partial volume dump/copy **RRG**
 selective restore **RRG**
 uses for **RRG**
 volume copy **RRG**
 volume dump **RRG**
 volume restore **RRG**
 core image library **HLPI**
 application program stored in **HLPI**
 DBDs stored in **HLPI**
 PSBs stored in **HLPI**
 correcting data bases **GNU**
 counters and clocks **DBA**
 creating secondary indexes **GNU**
 creating a data base, RPG II file description
 specification format **CALL**
 creating a secondary index **GNU**
 creating an online nucleus interactively **GNU**
 creating DL/I macros **IRDU**
 creating ISQL routine **IRDU**
 creating the target table **RDU**
 coding example **RDU**
 control statement (LI) **RDU**
 description **RDU**
 in ISQL Extract Defines Utility **RDU**
 input/output **RDU**
 job control statements **RDU**
 parameter statement **RDU**
 CSDA transaction (start MPS operation) **GNU**
 CSDB transaction (master partition controller) **GNU**
 CSDC transaction (batch partition controller) **GNU**
 CSDD transaction (stop MPS operation) **GNU**
 CSDE (CICS/VS) **DBA**
 CSDP transaction (purge temporary storage) **GNU**
 CSMT
 to detect missing BPCs **DIAG**
 CSSL (CICS/VS) **DBA**
 CSTT transaction (CICS/VS) **DBA**
 CUMCONST **DIAG**
 CUMIN **DIAG**
 current maximum task value **DBA**
 current position **CALL**
 and GN call **CALL**
 and segment insertion **CALL**
 current position in a data base **GNU**
 current roster
 analyzing associations **ABDD**
 combining local view **ABDD**
 defining application programs **ABDD**
 distributed vs local processing **ABDD**
 frequency of use **ABDD**
 grouping data elements with keys **ABDD**
 input requirements **ABDD**
 isolating duplicate values **ABDD**
 isolating repeating data elements **ABDD**
 local view as input **ABDD**
 mappings **ABDD**
 online vs batch **ABDD**
 primary keys **ABDD**
 report requirements **ABDD**
 requirement for **ABDD**
 CURRPOS (X'30') **DIAG**
 cursor position

- description **IRDU**
- customer data base **GNU**
 - DBD example phase **GNU**
 - load **PSB GNU**
 - load **PSB phase 3 GNU**
 - logical **PSB GNU**
 - logical **PSB phase 2 GNU**
 - physical **DBD phase 3 GNU**
 - PSB to process GNU**
 - sample application description **GNU**
 - sample record **GNU**
- customer file record layout **GNU**
- customer history segments (**STSCHIS**), description **GNU**
- customer location segment (**STSCLOC**), description **GNU**
- customer name and address segment (**STSCCST**), description **GNU**
- customer order segment (**STPCORD**), description **GNU**
- customer sample data base description **GNU**
- customer status segment (**STSCSTA**), description **GNU**

D

- D** command code **GNU**
- D** operand **GNU**
 - SEGM** statement in logical **DBD GNU**
- D** type online trace request **DIAG**
- DACT** (direct algorithm communication table) **DBA**
- DASD** types **DBA**
 - changing **DBA**
 - procedure **DBA**
- data access, controlling **DBA**
- data administration **GIM**
- data aggregate
 - class **ADBD**
 - correspondence to segment **ADBD**
 - definition **ADBD**
- data areas, **MPS DIAG**
- data base **GNU, HLPI**
 - (See also concepts and terminology)
 - access method **GNU**
 - access methods, selecting **GNU**
 - access paths **GNU**
 - access techniques **DBA**
 - advantages of **GIM**
 - analysis, user responsibility **GNU**
 - and its secondary index **GNU**
 - as conceptual structure **ADBD**
 - attribute, data element **GNU**
 - backout utility **GNU**
 - change accumulation utility **GNU**
 - characteristics **HLPI**
 - checklist, data base design **GNU**
 - compared to conventional file **ADBD**
 - concepts **GNU**
 - contention, online, minimizing **GNU**
 - contents of **GNU**
 - correction **GNU**
 - data element **GNU**
 - data set image copy utility **GNU**
 - data set recovery utility **GNU**
 - defined **DBA**
 - defining physical characteristics **DBA**
 - defining structures **DBA**
 - flat files **DBA**
 - hierarchical structures **DBA**
 - defining **VSAM** clusters **GNU**
 - definitions **ADBD, GNU, HLPI**
 - using documentation aid **GNU**
 - description (**DBD**) **GNU**
 - logical **GNU**
 - physical **GNU**
 - description generation (**DBDGEN**) **GNU**
 - description of **DL/I HLPI**
 - design **DBA, GNU**
 - design and performance evaluation **GNU**
 - design checklist **GNU**
 - design concepts **GNU**
 - design objectives **GNU**
 - design process **GNU**
 - design tasks **GNU**
 - designing the application data structure **GNU**
 - designing the physical data structure **GNU**
 - DL/I** structural elements **DBA**
 - fields and segments **DBA**
 - segment types **DBA**
 - entities **GNU**
 - error detection **GNU**
 - extended structures **DBA**
 - logical child segment **DBA**
 - logical parent segment **DBA**
 - physical parent **DBA**
 - facility **GNU**
 - gathering requirements, data base design **GNU**
 - hierarchical data structure **GNU**
 - hierarchy **HLPI**
 - implementation **GNU**
 - Gantt chart **GNU**
 - gross **PERT** chart **GNU**
 - overview **GNU**
 - project approach **GNU**
 - project cycle **GNU**
 - sample project plan **GNU**
 - implementation compared to conventional file **ADBD**
 - implementing design of **DBA**
 - initial load/reload **GNU**
 - with logical relationships **GNU**
 - with secondary indexes **GNU**
 - label information **GNU**
 - entering **IRDU**
 - IUG GNU**
 - load **GNU**
 - IUG GNU**
 - processing **GNU**
 - with logical relationships **GNU**
 - with secondary indexes **GNU**
 - loading **DBA, GNU**
 - log **GNU**
 - log print utility **GNU**
 - logging facility **GNU**
 - logical **GNU**
 - management
 - through **DL/I GIM**
 - user responsibility **GNU**
 - monitoring **DBA**
 - online system **GNU**
 - operations, user responsibility **GNU**
 - partial reorganization (**DLZPRCT1** and **DLZPRCT2**) **GNU**
 - performance aspects, physical data structure **GNU**
 - physical **GNU**
 - physical implementation of conceptual structure **ADBD**
 - positioning **GNU**
 - prefix resolution utility (**DLZURG10**) **GNU**
 - prefix update utility (**DLZURGP0**) **GNU**
 - preorganization utility (**DLZURPR0**) **GNU**
 - processing functions, application program **GNU**
 - processing methods **HLPI**
 - processing vs conventional file processing **ADBD**
 - PSBs**, logical **GNU**
 - randomizing modules **DBA**

- record **GNU**
 - access path **GNU**
 - description **GNU**
 - in physical storage **GNU**
 - key **GNU**
 - key field **GNU**
 - sequence field **GNU**
- record in physical storage **GNU**
- records **HLPI**
- recovery **DBA, GNU**
 - IUG **GNU**
- recovery plan **DBA**
- reorganization **GNU**
 - HD **GNU**
 - IUG **GNU**
- reorganization/load processing **GNU**
- reorganizing **DBA**
- request **DBA**
- restart **GNU**
- rules for data base structures **GNU**
- scan utility (DLZURGS0) **GNU**
- secondary index **GNU**
- segment grouping, data base design **GNU**
- segment, number of segment types **HLPI**
- space
 - acquiring **DBA**
 - freeing **DBA**
- structure **GNU**
 - hierarchical **GNU**
 - physical **GNU**
- structures **DBA**
- synonyms **GNU**
- transaction **GNU**
- transaction/data element matrix **GNU**
- utilities **DBA**
- views of **HLPI**
- views of data **DBA**
 - application **DBA**
 - system **DBA**
- data base access methods **DBA, DIAG**
- data base administration
 - choice of organization and access method **ADBD**
 - controlling operation **DBA**
 - data administration and security **DBA**
 - improving performance **DBA**
 - monitoring **DBA**
 - recovery **DBA**
- decisions **ADBD**
- definition **ADBD**
- function **DBA**
- implementation requirements and data base
 - design decisions **ADBD**
- implementing design **DBA**
- installation planning **DBA**
 - pre-installation **DBA**
- link to application design **ADBD**
- physical implementation **ADBD**
- summary chart **DBA**
- task of physical data base design **ADBD**
- tasks **DBA**
- using data bases **DBA**
 - DL/I program execution **DBA**
 - loading data bases **DBA**
 - modifying data bases **DBA**
- data base administration (see DBA) **HLPI**
- data base administrator, responsibilities of **CALL**
- data base back-up **ADBD**
- data base backout utility **DIAG, RDU, RRG**
 - control statements **IRDU**
 - description **IRDU**
 - illustration **IRDU**
 - input to **IRDU, RRG**
 - log file **RRG**
 - output from **IRDU, RRG**
 - parameters **IRDU**
- data base buffer pool and control blocks **DBA**
- data base buffering **DBA**
- data base call tracing (DLZTRACE) **DIAG**
- data base catalog entry **DIAG**
- data base change accumulation utility **DIAG, RDU**
 - coding examples **RDU**
 - control statements **IRDU, RDU**
 - AI **RDU**
 - AO **RDU**
 - DB0 **RDU**
 - DB1 **RDU**
 - ID **RDU**
 - LI **RDU**
 - LO **RDU**
 - description **IRDU, RDU**
 - illustration **IRDU**
 - input/output **IRDU, RDU**
 - job control statements **RDU**
 - parameters **IRDU**
- data base concepts **CALL**
- data base considerations **DBA**
- data base data set image copy
 - control statements **IRDU**
 - description **IRDU**
 - illustration **IRDU**
 - input **IRDU**
 - output **IRDU**
 - parameters **IRDU**
- data base data set image copy utility **RDU**
 - coding examples **RDU**
 - control statement (D) **RDU**
 - control statements **RDU**
 - description **RDU**
 - input/output **RDU**
- data base data set recovery utility **DIAG, RDU**
 - coding examples **RDU**
 - control statements **IRDU, RDU**
 - DI **RDU**
 - LI **RDU**
 - S **RDU**
 - description **RDU**
 - illustration **IRDU**
 - input/output **IRDU, RDU**
 - job control statements **RDU**
 - parameter statement **RDU**
 - parameters **IRDU**
- data base definitions **DBA, RDU**
 - documenting **RDU**
 - storing in SQL/DS tables **DBA**
- data base deletions **CALL**
- data base description (DBD) **CALL, HLPI, LLC**
 - (see also control data base initialization) **LLC**
 - control data base **LLC**
 - HDAM used for **LLC**
 - HIDAM used for **LLC**
 - restriction on change of segment layout or names **LLC**
 - segment types **LLC**
 - LLCTL **LLC**
 - LLCTL, layout **LLC**
 - PARTBEXP **LLC**
 - PARTBEXP, layout **LLC**
 - UPDMASTER **LLC**
 - UPDMASTER, layout **LLC**
 - structure **LLC**

use **LLC**
 data base description generation (see **DBDGEN**) **IRDU, RDU**
 data base design **DBA**
 benefits of good **ADBD**
 consequences of poor **ADBD**
 creating local views **ADBD**
 objectives **ADBD**
 process **ADBD**
 tools **ADBD**
 Data Base Design Aid **ADBD**
 DB/DC Data Dictionary **ADBD**
 Documentation Aid **ADBD**
 Data Base Design Aid **ADBD**
 data base displaying **DIAG**
 data base image copy utility **DIAG**
 data base label information
 control statements **IRDU**
 description **IRDU**
 parameters **IRDU**
 data base load **CALL**
 control statements **IRDU**
 description **IRDU**
 IRDU
 data base loading **CALL, DIAG, CALL**
 data base organization **DIAG**
 data base organizing and maintenance processor **LLC**
 data base physical characteristics
 allocating data sets **DBA**
 VSAM catalog **DBA**
 VSAM data sets **DBA**
 VSAM data spaces **DBA**
 CI and block size **DBA**
 device type **DBA**
 distributed free space **DBA**
 logical record length **DBA**
 scanning range **DBA**
 VSAM options **DBA**
 data base physical characteristics, defining **DBA**
 data base prefix resolution utility **DIAG, RDU, RRG**
 coding example **RDU**
 control statement (R) **RDU**
 description **IRDU, RDU**
 illustration **IRDU**
 input/output **IRDU, RDU**
 job control statements **RDU**
 parameters **IRDU**
 data base prefix update utility **DIAG, RDU**
 coding example **RDU**
 control statement (U) **IRDU, RDU**
 description **IRDU, RDU**
 illustration **IRDU**
 input/output **IRDU, RDU**
 job control statements **RDU**
 parameter statement **RDU**
 parameters **IRDU**
 data base preorganization utility **DIAG, RDU, RRG**
 coding example **RDU**
 control statements **IRDU, RDU**
 DBIL RDU
 DBR RDU
 OPTIONS RDU
 description **IRDU, RDU**
 illustration **IRDU**
 input/output **IRDU, RDU**
 job control statements **RDU**
 parameter statement **RDU**
 parameters **IRDU**
 data base processing **CALL**
 data base record **CALL, DBA, CALL**
 access techniques **DBA**

compared to physical record **ADBD**
 control by application program **DBA**
 control by order of creation **DBA**
 defining sequencing for **DBA**
 definition **ADBD, DBA**
 direct pointer sequencing **DBA**
 direct processing **DBA**
 direct access by randomizing routine **DBA**
 indexed direct access **DBA**
 sequential search direct access **DBA**
 example **DBA**
 functions **DBA**
 illustrated **DBA**
 indexed sequencing **DBA**
 levels in **ADBD**
 secondary indexes **DBA**
 alternate record entry point **DBA**
 sequence fields **DBA**
 sequential processing **DBA**
 subsequence fields **DBA**
 target segment **DBA**
 sequence of **CALL**
 sequential processing **DBA**
 sequential sequencing **DBA**
 data base recovery **DBA, DIAG**
 description **IRDU**
 data base recovery plan
 description **DBA**
 process **DBA**
 recovery and restart **DBA**
 application needs **DBA**
 documenting **DBA**
 program requirements **DBA**
 testing **DBA**
 types of failure **DBA**
 recovery facilities **DBA**
 recovery procedure flowchart **DBA**
 recovery-restart procedure **DBA**
 special problems **DBA**
 data base recovery utilities **DIAG, IRDU**
 backout **DIAG**
 change accumulation **DIAG**
 data set recovery **DIAG**
 image copy **DIAG**
 log print **DIAG**
 data base reorganization **DIAG**
 description **IRDU**
 utility sequence **IRDU**
 data base reorganization utilities **IRDU, RDU**
 HD reorganization reload **DIAG**
 HD reorganization unload **DIAG**
 HISAM reorganization reload **DIAG**
 HISAM reorganization unload **DIAG**
 partial data base reorganization **DIAG**
 data base requirements **DBA**
 data base retrievals **CALL**
 data base scan **DBA**
 data base scan utility **DIAG, RDU, RRG**
 coding example **RDU**
 control statements **IRDU, RDU**
 CHKPT RDU
 DBS RDU
 RSTRT RDU
 description **IRDU, RDU**
 illustration **IRDU**
 input/output **IRDU, RDU**
 job control statements **RDU**
 parameter statement **RDU**
 parameters **IRDU**
 data base space allocation **DBA**

data base structures **ADBD, DBA**
 considerations in defining **DBA**
 extended **DBA**
 flat files **DBA**
 hierarchical **DBA**
 elements **DBA**
 data base **DBA**
 field **DBA**
 record **DBA**
 segment **DBA**
 segment occurrence **DBA**
 segment type **DBA**
 flat file **ADBD**
 hierarchical **ADBD**
 network **ADBD**
 organizations **DBA**
 extended **DBA**
 flat files **DBA**
 hierarchical **DBA**
 data base updates **CALL**
 data base utilities
 backout utility - **DLZBACK0 DBA**
 change accumulation utility - **DLZUCUM0 DBA**
 data set image copy utility - **DLZUDMP0 DBA**
 data set recovery utility - **DLZURDB0 DBA**
 prefix resolution utility - **DLZURG10 DBA**
 prefix update utility - **DLZURGP0 DBA**
 preorganization utility - **DLZURPR0 DBA**
 scan utility - **DLZURGS0 DBA**
 data base, loading **IRDU**
 data base, protecting your **DBA**
 data base, remote **DIAG**
 data base, reorganizing **IRDU**
 data bases, logical **DBA**
 data communications requirements **ADBD**
 data conversion **DIAG**
 data conversion, automatic **CALL**
 data definition language, DL/I **DBA**
 data delivery **GIM**
 data dictionary **DBA**
 as design tool **ADBD**
 DB/DC **ADBD**
 description **ADBD**
 definition **ADBD**
 knowledge about data elements **ADBD**
 use in implementation plan **ADBD**
 use in sample application **ADBD**
 user created **ADBD**
 data element
 correspondence to field **ADBD**
 included in local view **ADBD**
 knowledge of required **ADBD**
 values **ADBD**
 data elements **GNU**
 grouping into physical segments **GNU**
 data encryption **ADBD**
 data entry items
 description **IRDU**
 data entry panels
 description **IRDU**
 ELIAS migration **IRDU**
 IMF, sample **IRDU**
 IUG, sample
 description **IRDU**
 NEXT macro entry example **IRDU**
 relating ACT macros to **IRDU**
 relating DBD macros to **IRDU**
 relating PSB macros to **IRDU**
 sample **IRDU**
 WHAT NEXT entry example **IRDU**
 data entry panels, interactive facilities **GNU**
 data fields **HLPI**
 duplicate **DBA**
 data format conversion **GNU**
 automatic **GNU**
 unsupported **GNU**
 data format, hierarchical direct **DIAG**
 data formats **DBA**
 application views **DBA**
 field level sensitivity **DBA**
 segment sensitivity **DBA**
 system views **DBA**
 segment edit/compression **DBA**
 variable length segments **DBA**
 data formatting capabilities **DBA**
 field level sensitivity **DBA**
 segment edit/compression **DBA**
 segment sensitivity **DBA**
 variable length segments **DBA**
 data gathering, PSB **DIAG**
 data in a segment, changing the position of **DBA**
 data independence **GNU**
 data integrity **DBA**
 maintaining **DBA**
 use of **DBA**
 data location aids **DIAG**
 data management **GIM**
 reduced data redundancy **GIM**
 data integrity **GIM**
 data security **GIM**
 data recovery **GIM**
 data management block (DMB) **CALL, DBA, GNU, HLPI**
 data management support requirements **ADBD**
 DATA operand **GNU**
 SEGM statement in logical DBD **GNU**
 data organizations **GIM**
 hierarchical direct **GIM**
 advantages of hierarchical direct **GIM**
 direct access methods **GIM**
 hierarchical sequential **GIM**
 advantages of hierarchical sequential **GIM**
 sequential access methods **GIM**
 data requirements, sample **DBA**
 data set **GNU**
 image copy utility **GNU**
 recovery utility **GNU**
 VSAM **GNU**
 VSAM share option **GNU**
 data set filename in segment name feedback **CALL**
 Data Set Image Copy Backup **RRG**
 Data Set Image Copy utility **RRG**
 backing up with **RRG**
 input to **RRG**
 output from **RRG**
 restoring from **RRG**
 data set recovery utility **DIAG, RRG**
 description **IRDU**
 input to **RRG**
 output from **RRG**
 data sets, allocating **DBA**
 data sharing, limited **DBA**
 data structure **GNU**
 as data base record **ADBD**
 compared to physical record **ADBD**
 data system environment **GIM**
 data base management through DL/I **GIM**
 data communications **GIM**
 data administration **GIM**
 data delivery **GIM**

data structure (see data base hierarchy) **HLPI**
 data types (see TYPE operand) **RDU**
 data-oriented utilities **RRG**
 data
 in the DL/I buffer pool **DIAG**
 in the VSAM data area **DIAG**
 intersection **HLPI**
 two views of **DBA**
DATASET statement **GNU**
 BLOCK operand **GNU**
 DD1 operand **GNU**
 DEVICE operand **GNU**
 logical DBD **GNU**
 RECORD operand **GNU**
 SCAN operand **GNU**
 VSAM **LLC**
DATASET statement format **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 LOGICAL data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
DB space, acquiring **DBA**
DB/DC Data Dictionary **ADBD, DBA**
DBA **HLPI**
 (See also data base administration)
 functions **HLPI**
 information supplied by **HLPI**
DBD (data base description) **ADBD, CALL, GNU, RDU**
 assembler listing **DIAG**
 building **CALL**
 coding a logical relationship in physical **GNU**
 confirming the **DBDGEN** **DIAG**
 creating interactively **GNU**
 data gathering **DIAG**
 default table variables **DIAG**
 definition table variables **DIAG**
 generation **DIAG**
 assembler listing **DIAG**
 job stream example for **VSE** **DIAG**
 logical **GNU**
 module flow **DIAG**
 naming convention **ADBD, DBA**
 physical **GNU**
 storing information into SQL/DS tables **RDU**
 tables **DIAG**
 use to determine data base organization **GNU**
DBD default table **IRDU**
DBD definition **IRDU**
DBD definition table **IRDU**
DBD definitions list table **IRDU**
DBD example **GNU**
 phase 2 customer data base **GNU**
 phase 2 inventory data base **GNU**
DBD generation (**DBDGEN**) **DBA, IRDU**
 errors **DIAG**
DBD generation procedure **RDU**
 CMS macro file **IRDU**
 diagnostic messages **RDU**
 generation **IRDU**
 job control statements **RDU**
 output listing **RDU**
 VSE job stream **IRDU**
DBD macros relating to IMF panels **IRDU**
DBD name in PCB mask **CALL**
DBD statement format **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 LOGICAL data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
DBD statement, basic **GNU**
 ACCESS operand **GNU**
 CIANPT operand **GNU**
 HISAM parameter, ACCESS operand **GNU**
 HSAM parameter, ACCESS operand **GNU**
 IMSCOMP parameter, ACCESS operand **GNU**
 NAME operand **GNU**
 PRIMCI operand **GNU**
 RILIM operand **GNU**
 SHISAM parameter, ACCESS operand **GNU**
 SHSAM parameter, ACCESS operand **GNU**
DBD tables **DIAG**
DBD worksheets **IRDU**
DBD, description **HLPI**
DBDACCESSDATA **RDU**
 contents **RDU**
 description **RDU**
DBDACCESSDATA, description of **DBA**
DBDACCESS **RDU**
DBDBASIC **RDU**
DBDBASICDATA **RDU**
 contents **RDU**
 description **RDU**
DBDBASICDATA, description of **DBA**
DBDFIELD **RDU**
DBDFIELDDATA **RDU**
 contents **RDU**
 description **RDU**
DBDFIELDDATA, description of **DBA**
DBDGEN (data base description generation) **DBA, DIAG, GNU, IRDU, RDU**
 Assembler listing **DIAG**
 coding conventions **GNU**
 confirming **DBD** generation **DIAG**
 control statement sequence **DBA**
 definition **IRDU, RDU**
 description **IRDU**
 END statement **GNU**
 errors **DIAG**
 execution, job control **GNU**
 FINISH statement **GNU**
 for the phase 1 data bases **GNU**
 generation **IRDU**
 generation step **RDU**
 IMF **GNU**
 input deck structure **GNU**
 logical relationships **GNU**
 macros **DBA**
 output listing, sample **RDU**
 secondary indexes **GNU**
 statement **GNU**
DBDGEN statement format **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**

LOGICAL data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 DBDGEN, IMF **GNU**
 DBDHIER **RDU**
 DBDLCHILDDATA **RDU**
 contents **RDU**
 description **DBA, RDU**
 DBDLCHLD **RDU**
 DBDNAME operand coding requirements **RDU**
 DBDNAME operand, PCB statement **GNU**
 dbdname parameter **RDU**
 DBDNAMES **RDU**
 DBDSEGM **RDU**
 DBDSEGMENTDATA **RDU**
 contents **RDU**
 description **DBA, RDU**
 DBOMP (see data base organizing and
 maintenance processor) **LLC**
 DBPCBDBD **DIAG**
 DBSPACE
 acquiring DBSPACE **DBA**
 DBD **DBA**
 estimating **DBA**
 PSB **DBA**
 running out of **DBA**
 specifications for **DBA**
 DB0 statement **RDU**
 DB1 statement **RDU**
 DCT (destination control table) **RDU**
 blocksize **DIAG**
 DDATA operand coding requirements **RDU**
 DDIRSYM **DIAG**
 DD1 operand coding requirements **RDU**
 DATASET statement **GNU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 DD2 operand coding requirements **RDU**
 HSAM data bases **RDU**
 SHSAM data bases **RDU**
 deadlock avoidance **DBA**
 program isolation **GNU**
 debugging (see problem determination) **CALL**
 aid - DLZDLTXX **DIAG**
 dumps **DIAG**
 execution time aids **HLPI**
 logical retrieve **DIAG**
 online ARSAabend **DIAG**
 online wait state **DIAG**
 your program **HLPI**
 decision tree chart **ADBD**
 default table
 ACT **DIAG, IRDU**
 DBD **IRDU**
 DBD variables **DIAG**
 IUG created **DIAG**
 PSB **IRDU**
 PSB variables **DIAG**
 default values
 description **IRDU**
 DEFINE command **CALL, GNU, RRG**

 access method services **GNU**
 use of **GNU**
 DEFINE run after DELETE run (VSAM) **LLC**
 defining **GNU**
 a DBD **IRDU**
 a PSB **IRDU**
 an ACT **IRDU**
 applications under CICS/VS (PPT) **GNU**
 data bases as DB-files (RPG II) **GNU**
 the online environment **GNU**
 the view of the data base **GNU**
 transactions (PCT) **GNU**
 VSAM clusters **GNU**
 VSAM data sets **GNU**
 defining data sets to VSAM **RDU**
 defining programs **ADBD**
 example **ADBD**
 requirements and considerations **ADBD**
 data base vs conventional file processing **ADBD**
 distributed vs local processing **ADBD**
 frequency of use **ADBD**
 input requirements **ADBD**
 online vs batch **ADBD**
 output requirements **ADBD**
 resource requirements **ADBD**
 type of access **ADBD**
 type of processing **ADBD**
 defining smaller logical units of work **RRG**
 defining structures **DBA**
 defining the data base **GIM**
 DBD (Data Base Description) **GIM**
 PSB (Program Specification Block) **GIM**
 DL/I Documentation Aid **GIM**
 Interactive Macro Facility (IMF) **GIM**
 definition of DBDs, PSBs, and ACBs **LLC**
 rules for names of **LLC**
 sample DBDs for control data base **LLC**
 sample DBDs for parts data base **LLC**
 sample PSBs for **LLC**
 control data base **LLC**
 executing Assembler application program **LLC**
 SEGM statement **LLC**
 BYTES parameter of **LLC**
 RULES parameter of **LLC**
 definition table
 ACT **DIAG, IRDU**
 DBD **IRDU**
 DBD variables **DIAG**
 PSB **IRDU**
 PSB variables **DIAG**
 definition, data base **GNU**
 definitional capabilities, choosing DL/I **DBA**
 delete **GNU**
 byte **GNU**
 rule for logical child segment **GNU**
 rule for physical parent segment **GNU**
 delete - DLET **DIAG**
 DELETE command **GNU, HLPI**
 access method services **GNU**
 format of **GNU**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 use with Get command **GNU**
 VSAM **RRG**
 delete data **GNU**

IUG GNU
DELETE function **CALL**
 definition of **CALL**
 example of **CALL**
 delete rules **DBA, GNU**
 for logical child segment **DBA, GNU**
 for logical parent segment **DBA, GNU**
 for physical parent segment **GNU**
DELETE run before **DEFINE** run (**VSAM**) **LLC**
 description **IRDU**
 delete/replace call path **DIAG**
 delete, warning **HLPI**
 deleting segments **DBA, GNU, HLPI**
 from an **HD** data base **GNU**
 in a data base **GNU**
 with a secondary index **GNU**
 deletion rules (see **RULES** operand) **GNU, RDU**
 deletions, updating of **LLC**
 not supported **LLC**
 delimiter, command **GNU, HLPI**
 definition **HLPI**
 description **HLPI**
 examples **HLPI**
 in syntax **HLPI**
 dependent child **HLPI**
 dependent segment **CALL, DBA, CALL, HLPI**
 definition **ADBD, CALL**
 deletion of **CALL**
 example **DBA**
 in data base record **ADBD**
 key fields **CALL**
 dependent segments, defining sequencing for **DBA**
 depends in logical relationships **DBA**
 describing an application program's view of a data base **RDU**
 describing characteristics of **DL/I** data bases **RDU**
 describing **DL/I** tables for an online system **RDU**
 description of **DL/I GNU**
 description of **PSBGEN** output **GNU**
 description of the online nucleus generation output **GNU**
 design **GNU**
 data base **GNU**
 process for data bases **GNU**
 design rules, logical relationship **GNU**
 design, data base **DBA**
 destination control table (see **DCT**) **RDU**
 destination parent **GNU**
 concatenated key (**DPCK**) **GNU**
 segment **HLPI**
 physical replace rule **CALL**
DEVADDR operand coding requirements **RDU**
HSAM data bases **RDU**
SHSAM data bases **RDU**
 device independence **GNU**
DEVICE operand coding requirements **RDU**
HD data bases **RDU**
HDAM data bases **RDU**
HIDAM data bases **RDU**
HISAM data bases **RDU**
HSAM data bases **RDU**
 primary **INDEX** data bases **RDU**
 secondary **INDEX** data bases **RDU**
SHISAM data bases **RDU**
SHSAM data bases **RDU**
DEVICE operand, **DATASET** statement **GNU**
 device types (see **DEVICE** operand) **DBA, RDU**
DFHALT DBA
DFHFCT - **CICS/VS** file control table **GNU**
DFHISP (**ISC** interface module) **DIAG**
DFHMIR, **CICS/VS** mirror program **DIAG**
DFHPCT - **CICS/VS** program control table **GNU**

DFHPCT example for **MDLI DIAG**
DFHPPT - **CICS/VS** processing program table **GNU**
DFHPPT examples for **MDLI DIAG**
DFHSG DBA
DI statement **RDU**
 Diagnostic Facility, **CICS/VS** Execution (**EDF**) **HLPI**
 diagnostic information, **PL/I HLPI**
 dialogue forms, interactive facilities **GNU**
DIB (**DL/I** Interface Block) **GNU, HLPI**
 description **HLPI**
 use of **HLPI**
 variables **GNU**
 for **COBOL HLPI**
 for **PL/I HLPI**
DIBFLAG HLPI
DIBKFBL GNU, HLPI
DIBSEGLV HLPI
DIBSEGM HLPI
DIBSTAT HLPI
DIBVER HLPI
 differences between batch, **MPS**, and online **DL/I GNU**
 direct access **ADBD**
 by direct pointer/sequential search **DBA**
 by randomizing routine **DBA**
 indexed **DBA**
 sequential search **DBA**
 direct access pointers in **HD GNU**
 direct access program **GNU**
 direct algorithm communication table (**DACT**) **DBA**
 direct organization **ADBD, DIAG**
 advantages **ADBD**
 definition **ADBD**
 disadvantage **ADBD**
 indexed access **ADBD**
 pointers **ADBD**
 randomized access **ADBD**
 direct organization access methods **DBA**
 indexed access **DBA**
 randomized access **DBA**
 direct pointer
 method **DBA**
 options **DBA**
 sequencing **DBA**
 direct processing **HLPI**
 direct processing of records **DBA**
 directed graphs **LLC**
 display data **GNU**
IUG GNU
 display table data
 description **IRDU**
 displaying
 a data base **DIAG**
ISQL routine **RDU**
SQL/DS table data **RDU**
 distributed data **DBA**
CICS/VS intersystem communication support **DBA**
 extended remote **PSB DBA**
 definition **DBA**
 operation **DBA**
 uses **DBA**
 remote **PSB DBA**
 centralized data base in two system configuration **DBA**
 definition **DBA**
 mirror transaction **DBA**
 operation **DBA**
 programming notes **DBA**
PSB names included **DBA**
 separate data bases in two system configuration **DBA**
 specifying **DBA**

- uses **DBA**
- distributed free space **DBA, GNU**
- distribution of LLC/CC in DL/I **LLC**
- DL/I **HLPI, RDU**
 - characteristics of data bases **ADBD**
 - commands **HLPI**
 - data bases **HLPI**
 - environments **HLPI**
 - extracting data to SQL/DS **RDU**
 - functions **HLPI**
 - preparing to use **HLPI**
 - programming techniques and suggestions **HLPI**
 - response to commands **HLPI**
 - terms **ADBD**
- DL/I access methods
 - reasons for change **DBA**
 - things to consider **DBA**
 - HD indexed to HD randomized **DBA**
 - HD indexed to HISAM **DBA**
 - HD randomized to HD indexed **DBA**
 - HD randomized to HISAM **DBA**
 - HISAM to HD indexed **DBA**
 - HISAM to HD randomized **DBA**
- DL/I backout failure **RRG**
- DL/I batch nucleus and program request handler **DBA**
- DL/I batch system initialization **GNU**
- DL/I batch system storage **DBA**
 - example **DBA**
 - partition contents **DBA**
 - partition values required **DBA**
- DL/I bridge (see Chained File - DL/I Bridge) **LLC**
- DL/I calls
 - (See calls or commands)
- DL/I checkpoint **CALL**
 - for MPS restart facility **CALL**
- DL/I CICS/VS interface checking (see checking the response to a DL/I call in a CICS/VS environment) **CALL**
- DL/I commands
 - (See calls or commands)
- DL/I data **RDU**
 - extracting **RDU**
- DL/I data base buffer pool and control blocks **DBA**
- DL/I data base utilities storage requirements **DBA**
- DL/I data definition language **DBA**
 - data base description **DBA**
 - program specification block **DBA**
- DL/I data management blocks (DMBs) **DBA**
- DL/I definitional capabilities
 - application view **DBA**
 - additional specifications **DBA**
 - longest concatenated key length **DBA**
 - multiple positioning **DBA**
 - secondary index usage **DBA**
 - segment contents **DBA**
 - segment organization **DBA**
 - segment processing options **DBA**
 - specifying base data base **DBA**
 - specifying segments **DBA**
 - base segment **DBA**
 - for a concatenated segment **DBA**
 - for a simple segment **DBA**
 - bidirectional logical relationship **DBA**
 - logical child segment **DBA**
 - logical child segment format **DBA**
 - pointer options **DBA**
 - rules **DBA**
 - suppression of logical twin pointers **DBA**
 - data formats **DBA**
 - field level sensitivity **DBA**
 - segment edit/compression **DBA**
 - segment sensitivity **DBA**
 - variable length segments **DBA**
 - defining fields **DBA**
 - field data type **DBA**
 - field length **DBA**
 - specifying the segment **DBA**
 - defining indexes **DBA**
 - HD primary **DBA**
 - HD secondary **DBA**
 - HISAM and SHISAM **DBA**
 - defining indexes for HIDAM and HDAM **DBA**
 - HIDAM/HDAM secondary index **DBA**
 - index data base **DBA**
 - primary index specification **DBA**
 - direct pointer options **DBA**
 - physical child last pointer **DBA**
 - physical twin backward pointer **DBA**
 - suppression of physical twin pointers **DBA**
 - field location in segment **DBA**
 - index data base **DBA**
 - left-to-right ordering of children **DBA**
 - logical data base **DBA**
 - defining fields for **DBA**
 - defining segments for **DBA**
 - defining structures for **DBA**
 - normal logical child **DBA**
 - segment length **DBA**
 - segment name **DBA**
 - parent/child relationships **DBA**
 - physical structures **DBA**
 - additional direct pointer options **DBA**
 - connection technique **DBA**
 - processing controls **DBA**
 - processing options **DBA**
 - rules for, defining **DBA**
 - segment contents **DBA**
 - segment name **DBA**
 - segments **DBA**
 - segment length **DBA**
 - sequencing for data base records **DBA**
 - sequencing by field value **DBA**
 - sequencing by order of creation **DBA**
 - sequencing under application control **DBA**
 - sequencing for dependent segments **DBA**
 - by field value **DBA**
 - by order of creation **DBA**
 - under application control **DBA**
 - sequencing for logical relationships **DBA**
 - sequence of logical parents **DBA**
 - sequence of physical parents **DBA**
 - unidirectional logical relationships **DBA**
 - logical child segment **DBA**
 - logical child segment format **DBA**
 - virtual logical child **DBA**
- DL/I DSECT assembly **DIAG**
- DL/I dumps
 - batch and MPS batch application dumps **DIAG**
 - invoking batch and MPS batch application dumps **DIAG**
 - batch utility dumps **DIAG**
 - dump output processing **DIAG**
 - illustrated **DIAG**
 - kinds of **DIAG**
 - online formatted system dump **DIAG**
 - invoking an online formatted system dump **DIAG**
 - online formatted task dump **DIAG**
 - invoking an online formatted task dump **DIAG**
 - problem determination **DIAG**
 - types of failure **DIAG**
 - when taken **DIAG**
- DL/I functions and HLPI commands **GNU**

DL/I High Level Programming Interface (see DL/I HLPI) **HLPI**

DL/I HLPI **HLPI**

- commands **HLPI**
- description of **HLPI**
- functions **HLPI**
- introduction **HLPI**

DL/I initialization **GNU**

- JCL requirements, MPS **GNU**
- MPS batch partition **GNU**

DL/I initialization and termination routines **DBA**

DL/I Interface Block (see DIB) **HLPI**

DL/I log **RRG**

DL/I modules **DBA**

- DB organization dependent **DBA**
- storage estimate **DBA**

DL/I MPS system storage **DBA**

- DL/I nucleus (DLZNUCxx) **DIAG**

DL/I online nucleus **DBA**

DL/I online system storage **DBA**

DL/I operational capabilities

- online operation **DBA**
 - CICS/VS system generation requirements and options **DBA**
 - online nucleus generation requirements **DBA**

DL/I parameter information **DIAG**

- for application programs **DIAG**
- for data base reorganization **DIAG**
- for data set recovery **DIAG**
- for HD unload utility **DIAG**
- for logical relationship resolution **DIAG**

DL/I parameter requirements **RDU**

DL/I partition specification table **DBA**

DL/I PRH to CICS/VS ISC interface **DIAG**

DL/I program specification block **DBA**

DL/I real storage estimates **DBA**

- application programs **DBA**
- DL/I action modules **DBA**
- DL/I buffer pool **DBA**
- DL/I control blocks **DBA**
- example **DBA**
- VSAM action modules **DBA**
- VSAM buffers **DBA**
- VSAM control blocks **DBA**

DL/I recovery utilities

- data base backout **DBA**
 - illustrated **DBA**
- data base recovery **DBA**
 - description **DBA**
 - illustrated **DBA**
- DL/I-IMS compatible data bases **DBA**
 - recovery of **DBA**
- functions supported **DBA**
- program names **DBA**

DL/I recovery/restart

- VSAM considerations **DBA**
 - catalog **DBA**
 - closing data sets **DBA**
 - facilities **DBA**
 - minimizing damage **DBA**

DL/I requests **CALL**

- in Assembler program **CALL**
- in COBOL program **CALL**
- in PL/I program **CALL**

DL/I sample application **ADBD**

DL/I status codes **CALL**

DL/I strategies **RRG**

DL/I structural organizations **DBA**

- extended structures **DBA**
- flat files **DBA**
- hierarchical structures **DBA**
 - child segment **DBA**
 - parent segment **DBA**
 - root segment **DBA**

DL/I system **DBA**

- access method modules used **DBA**
- major parts of **DBA**

DL/I system calls **CALL**

DL/I tables **DBA**

DL/I test program **DBA**

DL/I to CICS/VS ISC interface **DIAG**

DL/I trace facility **DIAG**

- defining the **DIAG**
 - DLZTRACE macro **DIAG**
 - DLZTRENT macro **DIAG**
- trace entry format **DIAG**
- trace types **DIAG**

DL/I trace, sample user exit routines **DIAG**

DL/I virtual storage estimates **DBA**

DL/I virtual storage requirements worksheet **DBA**

DLBL parameters, entering **IRDU**

DLET (delete) **CALL, DIAG, GNU**

- detailed description **CALL**
- effects of **CALL**
 - request **DIAG**

DLI function code **HLPI, RDU**

DLI parameter information **DIAG**

DLISTATS **RDU**

DLPV abend code **DBA**

DLR function code **HLPI, RDU**

DLZACT macro **GNU**

- BFRPOOL parameter **GNU**
- CMAXTSK parameter **GNU**
- DL/I online nucleus generation **GNU**
- HDBFR parameter **GNU**
- HSBFR parameter **GNU**
- MAXTSK parameter **GNU**
- PASS parameter **GNU**
- PGMNAME parameter **GNU**
- PI parameter **GNU**
- REMOTE parameter **GNU**
- SLC parameter **GNU**
- SUFFIX parameter **GNU**
- TYPE=BUFFER **DIAG**
- TYPE=CONFIG **DIAG**
- TYPE=FINAL **DIAG**
- TYPE=INITIAL **DIAG**
- TYPE=PROGRAM **DIAG, GNU**
- TYPE=RPSB **DIAG**

DLZACT statements **DBA**

- DLZACT TYPE=BUFFER statement format **RDU**
- DLZACT TYPE=CONFIG statement format **RDU**
- DLZACT TYPE=FINAL statement format **RDU**
- DLZACT TYPE=INITIAL statement format **RDU**
- DLZACT TYPE=PROGRAM statement format **RDU**
- DLZACT TYPE=RPSB statement format **RDU**
- DLZALTS, return code checking **DIAG**
- DLZBACK0 (see data base change accumulation utility) **DBA**
- DIAG, RDU**DLZBH
 - in DLZRLNK M **DIAG**
 - return code checking **DIAG**

DLZCBL10 **HLPI**

- COBOL HLPI sample program **GNU**

DLZCBL20, COBOL HLPI sample program **GNU**

DLZCBL30 **HLPI**

- COBOL HLPI sample program **GNU**
- GET command examples **HLPI**
- INSERT, REPLACE, DELETE command examples **HLPI**
- SCHEDULE, TERMINATE, CHECKPOINT
 - command examples **HLPI**

DLZCPY10 **DBA**

DLZDBH0n **DIAG**

DLZDBH00 DIAG
 linkage editor warning **DIAG**
DLZDDLE0 DBA
DLZDHDS0 DBA
DLZDLD00 DBA
 macros **DIAG**
DLZDLTXX test program DIAG
 call statement format **DIAG**
 comment statement format **DIAG**
 compare statement format (PCB compare) **DIAG**
 control statement formats **DIAG**
 example for loading a test data base **DIAG**
 general description **DIAG**
 other special statements **DIAG**
 PI support testing **DIAG**
 punch statement format **DIAG**
 sample job control statements **DIAG**
 Special call statements **DIAG**
 status statement format **DIAG**
 suggestions for using **DIAG**
 to display a data base **DIAG**
 to find position of last record in output data base **DIAG**
 to load a data base **DIAG**
 to perform regression testing **DIAG**
 to verify how a call is executed **DIAG**
 to verify input data base **DIAG**
 used as a debugging aid **DIAG**
DLZDSEH0 DIAG
DLZFSDP0 formatted system dump program GNU
DLZLI000 (language interface module) DIAG
DLZLOGP0 DBA, DIAG
 (see log print utility) **RDU**
DLZLI1000 (see language interface module) CALL
DLZMDLI0 - online test program DBA, DIAG
DLZMINIT DIAG
DLZMPC00 DIAG
DLZMPI00 DIAG
 DLZMABND - MPS batch abend routine **DIAG**
 DLZMINIT - MPS batch initialization **DIAG**
 DLZMMSG - MPS batch message writer **DIAG**
 DLZMPRH - MPS batch program request handler **DIAG**
 DLZMTERM - MPS batch termination **DIAG**
DLZNUCXX DIAG
 linkage editor warning **DIAG**
DLZNUCxx - DL/I nucleus DIAG
DLZODP DIAG
 linkage editor warning **DIAG**
DLZOLI00 DIAG
DLZOVSEX, VSAM asynchronous exit routine DIAG
DLZPLI10 HLPI
 PL/I HLPI sample program **GNU**
DLZPLI20, PL/I HLPI sample program GNU
DLZPLI30 HLPI
 GET command examples **HLPI**
 INSERT, REPLACE, DELETE command examples **HLPI**
 PL/I HLPI sample program **GNU**
 SCHEDULE, TERMINATE, CHECKPOINT
 command examples **HLPI**
DLZPRCT1 (partial reorganization) GNU
 (see partial data base reorganization utility, part 1) **RDU**
DLZPRCT2 (partial reorganization) GNU
 (see partial data base reorganization utility, part 2) **RDU**
DLZPRH00 DIAG
DLZQUEF0 DBA
DLZRCLL macro DIAG
DLZRDBC0 DIAG
DLZRDBL0 DBA
DLZREXT macro DIAG
DLZRHDR macro DIAG
DLZRLNK C macro DIAG
DLZRLNK D macro DIAG
DLZRLNK M macro DIAG
DLZRRRC00 DIAG
DLZRTL macro DIAG
DLZSAMCP, sample compression/expansion routine GNU
DLZSAM40 sample load program GNU
DLZSAM50 sample batch print program GNU
DLZSAM60 online sample application program GNU
 CICS/VS-DL/I tables **GNU**
 DFHFCT-CICS/VS file control table **GNU**
 DFHPCT-CICS/VS program control table **GNU**
 DFHPPT-CICS/VS processing program table **GNU**
 DLZACT-DL/I online nucleus generation **GNU**
 job control **GNU**
 screen formats **GNU**
DLZSLC TYPE=ENTRY statement format RDU
DLZSLC TYPE=FINAL statement format RDU
DLZSLC TYPE=INITIAL statement format RDU
DLZTPRT0 trace print utility DIAG
DLZTRACE macro DIAG
 CALLCON operand **DIAG**
 OPTION operand **DIAG**
 options and parameters **DIAG**
 OUTPUT operand **DIAG**
 sample trace user exit routines **DIAG**
 STOPKEY operand **DIAG**
 STRTKEY operand **DIAG**
 TRACSIZ operand **DIAG**
 TRCECON operand **DIAG**
 TYPETRC operand **DIAG**
 user exit **DIAG**
 USREXIT operand **DIAG**
DLZTRACE macro options DIAG
 BHINTF Type 1 (X'61') **DIAG**
 BHINTF Type 2 (X'62') **DIAG**
 CURRPOS (X'30') **DIAG**
 INDEXTRC (X'70') **DIAG**
 MODTRACE (X'10') **DIAG**
 ONLINEBH (X'80') **DIAG**
 PITRACE (X'90') **DIAG**
 RETRIEVE (X'20') **DIAG**
 USERCALL Type 1 (X'01') **DIAG**
 USERCALL Type 2 (X'02') **DIAG**
 VSAMINTF (X'50') **DIAG**
DLZTRCAL trace invocation macro DIAG
 operands **DIAG**
 parameter list **DIAG**
DLZTRENT macro DIAG
DLZTXIT0
 module description **DIAG**
 sample trace user exit routines **DIAG**
**DLZUACB0, application control blocks creation and
 maintenance DBA, GNU**
 BUILD statement **GNU**
 control statements **GNU**
 job control **GNU**
DLZUCUM0 DBA, DIAG
DLZUDMP0 DBA, DIAG
 (see data base data set image copy utility) **RDU**
DLZURDB0 DBA, DIAG
 (see data base data set recovery utility) **RDU**
DLZURGL0 (HD reorganization reload) utility DBA, DIAG, GNU
 (see HD reorganization reload utility) **RDU**
DLZURGP0 (data base prefix update utility) DBA, DIAG, GNU
 (see data base prefix update utility) **RDU**
DLZURGS0 (data base scan utility) DBA, DIAG, GNU
 (see data base scan utility) **RDU**
DLZURGU0 (HD reorganization unload utility) DBA, DIAG, GNU
 (see HD reorganization unload utility) **RDU**
DLZURG10 (data base prefix resolution utility) DBA, DIAG, GNU

(see data base prefix resolution utility) **RDU**
DLZURPR0 (data base prereorganization utility) **DBA, DIAG, GNU**
DLZURPR0 (see data base prereorganization utility) **RDU**
DLZURRL0 (HISAM reorganization reload utility) **DBA, DIAG, GNU**
 (see HISAM reorganization reload utility) **RDU**
DLZURUL0 (HISAM reorganization unload utility) **DBA, DIAG, GNU**
 (see HISAM reorganization unload utility) **RDU**
DLZXMT0 DBA
DLZ020I message **GNU**
DLZ031I message **GNU**
DLZ105I message **GNU**
DLZ106I message **GNU**
DLZ108I message **GNU**
DLZ113I message **GNU**
DMB (data management block) **DBA, GNU, HLPI**
 assignments **GNU**
 deleting from core image library **DIAG**
 in figure **HLPI**
 locating **DIAG**
 operand, **BUILD** statement (**PSBGEN**) **GNU**
 rebuilding **DIAG**
DMB operand **RDU**
 of **BUILD** control statement **RDU**
 Documentation Aid **ADBD, DBA, DIAG, GNU, RDU**
 accessing data in **SQL/DS** tables **RDU**
 acquiring **DBSPACE DBA**
 as used for **ISQL EXTRACT DEFINES** **RDU**
 data base definitions **DBA**
 description **RDU**
 example of storage requirements **DBA**
 installation **RDU**
 invoking during **ACBGEN DBA**
ISQL query routines **RDU**
 job control statements **RDU**
 job streams **RDU**
 modifying **RDU**
 retrieving **RDU**
 monitoring data definitions **DBA**
 specifying **GNU**
 tables **DBA**
 using **GNU, RDU**
 documenting data base definitions **RDU**
 dope vector tables **DIAG**
 double arrow link
 example **ADBD**
 isolated attributes **ADBD**
 secondary key candidates **ADBD**
DPCK (destination parent concatenated key) **GNU**
DSA (dynamic storage area) **DIAG**
DSECT assembly **DIAG**
 errors **DIAG**
DSECT generation for trace entries **DIAG**
 dump control, **DL/I** **DIAG**
 dump interpretation
 after an **abend** message **DIAG**
 dump output processing **DIAG**
 dump program, **DLZFSDP0 GNU**
DUMPIN **DIAG**
 dumps (see **DL/I** dumps)
 duplicate data fields (see **DDATA** operand) **RDU**
 dynamic segment expansion **GNU**
 dynamic storage area (**DSA**) **DIAG**
 Dynamic Transaction Backout utility (**DTB**) **GNU, RRG**
 integrity **GNU**
 dynamically scheduling execution **GNU**

E

EDF (see Execution Diagnostic Facility) **HLPI**
 edit/compression, segment **DBA**
 editing segments **GNU**
 education company
 applications **ADBD**
 class data aggregate **ADBD**
 class schedule **ADBD**
 current roster **ADBD**
 education centers **ADBD**
 headquarters **ADBD**
 instructor schedules **ADBD**
 instructor skills **ADBD**
 logical relationships **ADBD**
 secondary indexing **ADBD**
ELEM statement **GNU**
 for **CHKPID** option **GNU**
 for **FROM | INTO** option **GNU**
 for **PCB** option **GNU**
 format of **GNU**
SSA GNU
SSALIST option format **GNU**
 elements of the command language **HLPI**
ELIAS migration
 description **IRDU**
 indexes **IRDU**
 to **IMF** **IRDU**
 using **IRDU**
ELIAS table migration **DIAG**
ELIST command **GNU**
 format of **GNU**
 ellipses, coding convention **HLPI**
 emergency restarts **RDU**
 MPS restart facility **RDU**
 encryption **DBA**
 of data **ADBD**
 segment edit/compression routine **DBA**
 end item **LLC**
 code of zero **LLC**
 defined **LLC**
END statement format **RDU**
HD data bases **RDU**
HDAM data bases **RDU**
HIDAM data bases **RDU**
HISAM data bases **RDU**
HSAM data bases **RDU**
LOGICAL data bases **RDU**
 primary **INDEX** data bases **RDU**
PSB definition **RDU**
 secondary **INDEX** data bases **RDU**
SHISAM data bases **RDU**
SHSAM data bases **RDU**
END statement in logical **DBD GNU**
END statement, **DBDGEN GNU**
 entering label information **IRDU**
 entities **GNU**
 entry **ID** field, online trace **DIAG**
 entry sequenced data set (**ESDS**) **GNU**
 two areas of **GNU**
 entry to an application program **GNU**
COBOL GNU
PL/I GNU
RPG II GNU
 entry to batch or **MPS** batch programs **HLPI**
 environments
 execution **GNU**
 operating **HLPI**
 error checking **HLPI**

error conditions in IMF **DIAG**
 error messages **IRDU, LLC**
 for abnormal termination **CALL**
 error recovery **LLC**
 causes of error **LLC**
 in initial generation mode **LLC**
 in update mode **LLC**
 return codes **LLC**
 system errors **LLC**
 error termination
 CICS/VS internal **DBA**
 DL/I internal **DBA**
 errors **HLPI**
 ACT generation **DIAG**
 application control block creation and maintenance **DIAG**
 Assembler language programs **DIAG**
 batch application program **DIAG**
 batch job control **DIAG**
 batch user **DIAG**
 DBD generation **DIAG**
 DSECT assembly **DIAG**
 execution **HLPI**
 HD randomizing module **DIAG**
 initialization **HLPI**
 job control **DIAG**
 module assembly **DIAG**
 open/close (DLZDLOC0) **DIAG**
 trace print utility (DLZTPRT0) **DIAG**
 MPS batch user **DIAG**
 MPS Restart **DIAG**
 online application program **DIAG**
 online job control **DIAG**
 online user **DIAG**
 operation **DIAG**
 randomizing module errors **DIAG**
 ESDS (entry sequenced data set) **GNU**
 two areas of **GNU**
 establishing logical relationships in physical data bases **RDU**
 establishing secondary indexing in physical data bases **RDU**
 examples **GNU, HLPI, LLC**
 application design with data bases **ADBD**
 call **LLC**
 Assembler language **LLC**
 COBOL **LLC**
 PL/I **LLC**
 defining the extrapartition dataset (DCT) for DLZTRACE **DIAG**
 exploding techniques **LLC**
 FIRST and LAST **HLPI**
 loading data bases **GNU**
 log write times **GNU**
 logical DBDs **GNU**
 of output **RDU**
 from ISQL DA run routines **RDU**
 physical DBDs **GNU**
 physical DBDs with logical relationships **GNU**
 prefix resolution utility **GNU**
 prefix update utility **GNU**
 preorganization utility **GNU**
 segment selection **HLPI**
 suppressing creation of secondary indexes, example **RDU**
 trace print utility **DIAG**
 exclusive intent **DBA**
 EXEC DLI **HLPI**
 HLPI trigger **GNU**
 EXEC statement **HLPI**
 batch **HLPI**
 entering **IRDU**
 MPS batch **HLPI**
 EXEC translator, CICS/VS **GNU, HLPI**
 EXECUTE DLI **HLPI**
 HLPI trigger **GNU**
 executing **GNU**
 CICS/VS with MPS **GNU**
 DL/I programs **RDU**
 ISQL query routine **RDU**
 MPS batch programs **GNU**
 using MPS restart **GNU**
 your program **HLPI**
 execution **GNU, HLPI**
 batch and MPS batch **HLPI**
 batch program **GNU**
 debugging aids **HLPI**
 errors **HLPI**
 job stream **HLPI**
 DLBL statement **HLPI**
 TLBL statement **HLPI**
 of concurrent DL/I tasks (ACT) **GNU**
 of DBDGEN, job control **GNU**
 of DL/I programs **GNU**
 of LLC/CC in DL/I **LLC**
 initialized via DLZRRRC00 **LLC**
 parameter statement **LLC**
 PROCOPT **LLC**
 sample **LLC**
 of PSBGEN, job control **GNU**
 online **HLPI**
 scheduling **GNU**
 Execution Diagnostic Facility, CICS/VS (EDF) **HLPI**
 existing fields, redefining **DBA**
 existing programs, use of **DBA**
 exit condition codes (online) **DIAG**
 exit conditions (online) **DIAG**
 exit routine, field **GNU**
 exit routine
 user field **CALL**
 (see SUPRTN and EXTRTN operands) **RDU**
 EXIT 1, DLZTRACE user exit **DIAG**
 EXIT 2, DLZTRACE user exit **DIAG**
 EXIT 3, DLZTRACE user exit **DIAG**
 explicit definition **DBA**
 exploding technique **LLC**
 example **LLC**
 explosion **LLC**
 sequence **LLC**
 expression **HLPI**
 extended call interface (see UIB) **CALL**
 extended remote PSB **DBA, DIAG**
 extended secondary index support **DBA**
 extended structures **DBA**
 features of **DBA**
 bidirectional relationships **DBA**
 bidirectional versus unidirectional relationships **DBA**
 concatenated segments **DBA**
 considerations in defining **DBA**
 delete rules **DBA**
 insert rules **DBA**
 intersection data **DBA**
 logical data bases **DBA**
 optional pointers for logical relationships **DBA**
 physical structure rules for logical children **DBA**
 physical versus logical relationships **DBA**
 placement of logical child **DBA**
 replace **DBA**
 extended structures, defining **DBA**
 extended structures, features of **DBA**
 EXTRACT command **RDU**
 in ISQL EXTRACT DEFINES utility **RDU**
 EXTRACT DEFINES **GNU**
 IUG **GNU**

Extract Defines routine **RDU**
 invoking **RDU**
 EXTRACT DEFINES utility **GNU, RDU**
 extract facility overview **RDU**
 extracting DL/I data to SQL/DS **RDU**
 extracting the DL/I data **RDU**
 extrapartition data set **DIAG**
 defining for DLZTRACE **DIAG**
 extrapartition dataset **DIAG**
 defining for DLZTRACE **DIAG**
 example **DIAG**
 opening for DLZTRACE **DIAG**
 EXTRTN operand coding requirements **RDU**

F

F command code **GNU**
 facility, data base **GNU**
 failure during emergency restart **RRG**
 fast backup copy, utilities for **RRG**
 fast copy disk **RRG**
 functional support **RRG**
 uses for **RRG**
 volume copy **RRG**
 volume dump **RRG**
 volume restore **RRG**
 Fast Copy Disk utility **RDU**
 Fast Copy Disk/Copy Disk Volume Backup **RRG**
 FBA support **DIAG**
 FCT (file control table), CICS/VS **DBA, GNU, RDU**
 FEEDBACKLEN **HLPI**
 field correspondence to data element **ABDD**
 field data type, defining **DBA**
 field exit routine **GNU**
 defining a **DBA**
 user **CALL**
 field length, defining **DBA**
 field level sensitivity **CALL, DBA, DIAG, CALL, GNU**
 automatic data conversion **CALL**
 considerations **DBA**
 dynamic segment expansion **CALL**
 key feedback area **CALL**
 SSAs **CALL**
 status codes **CALL**
 user field exit routine **CALL**
 virtual fields **CALL**
 field level sensitivity debugging **DIAG**
 field level sensitivity parameter list **DIAG**
 field location in segment, defining **DBA**
 field name in qualified SSA **CALL**
 field name, defining the **DBA**
 FIELD statement **GNU**
 BYTES operand **GNU**
 NAME operand **GNU**
 SEQ parameter, NAME operand **GNU**
 START operand **GNU**
 TYPE operand **GNU**
 FIELD statement format **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 system-related fields **RDU**

/CK **RDU**
 /SX **RDU**
 field type, defining **DBA**
 field value for HD, sequencing by **DBA**
 field value for non-HD, sequencing by **DBA**
 field value, control by **DBA**
 field, maximum length **HLPI**
 field, virtual **GNU**
 FIELDLENGTH option **HLPI**
 in COBOL program **HLPI**
 in PL/I program **HLPI**
 fields **DBA, HLPI**
 and segments **DBA**
 and subfields, field level sensitivity **GNU**
 defining **DBA**
 of data **CALL**
 subsequence **DBA**
 virtual **CALL**
 file
 conventional **ABDD**
 compared to data base **ABDD**
 implementation compared to data base **ABDD**
 physical record **ABDD**
 processing vs data base processing **ABDD**
 flat **ABDD**
 file control table (FCT), CICS/VS **DBA, GNU, RDU**
 file definition, RPG II data base **GNU**
 file description specification **GNU**
 continuation line **GNU**
 format of **GNU**
 file integrity and recovery **GNU**
 file record layout, data base **GNU**
 file-oriented facilities **RRG**
 Change Accumulation utility **RRG**
 Data Set Image Copy utility **RRG**
 Data Set Recovery utility **RRG**
 financial industry **GIM**
 FINISH statement **GNU**
 DBDGEN **GNU**
 in logical DBD **GNU**
 FINISH statement format **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 LOGICAL data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 FIRST option **HLPI**
 fixed block architecture support **DIAG**
 fixed length segments **HLPI**
 (see BYTES operand) **RDU**
 flat file **ABDD**
 as data base **ABDD**
 definition **ABDD**
 flat files **DBA**
 format of scheduling command **GNU**
 format, basic DBDGEN control statements **GNU**
 formatted system dump program (DLZFSDP0) **GNU**
 forward recovery **RRG**
 Change Accumulation utility **RRG**
 Data Set Recovery utility **RRG**
 definition of **RRG**
 failure during **RRG**
 for previously successful job steps **RRG**
 from a data set image copy **RRG**

from a HISAM reorganization unload backup **RRG**
 from a restored data base **RRG**
 to a failure point and back out **RRG**
 utilities used **RRG**
 forward recovery failure **RRG**
 forward recovery log from backout utility **RRG**
 forward recovery with no backout **RRG**
 free space (see FRSPC operand) **GNU, RDU**
 anchor point **DIAG**
 chain pointer **DIAG**
 distributed **DBA, GNU**
 element **DIAG**
 freeing data base space **DBA**
 frequent backups **RRG**
 appropriate uses for **RRG**
 disadvantages of **RRG**
 FROM option **GNU, HLPI**
 path call **GNU**
 FROM|INTO option **GNU**
 FRSPC operand coding requirements **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 FRSPC operand, DATASET statement **GNU**
 function **HLPI**
 definition **HLPI**
 description **HLPI**
 example **HLPI**
 in syntax **HLPI**
 function code **CALL, HLPI, RDU**
 DLI **HLPI**
 DLR **HLPI**
 for MPS restart facility **CALL, RDU**
 function codes **DIAG**
 function names **GNU**
 CHKP (checkpoint) **GNU**
 DLET (delete) **GNU**
 GHN (get hold next) **GNU**
 GHNP (get hold next within parent) **GNU**
 GHU (get hold unique) **GNU**
 GN (get next) **GNU**
 GNP (get next within parent) **GNU**
 GU (get unique) **GNU**
 in RQDLI command **GNU**
 ISRT (insert) **GNU**
 PCB (schedule a PSB) **GNU**
 REPL (replace) **GNU**
 TERM (release a PSB) **GNU**
 functional restrictions, sequence fields **DBA**
 functions DL/I performs on the segment **HLPI**
 functions for access methods (SHISAM and HISAM) **GNU**
 functions, interactive **IRDU**

G

Gantt chart **DBA**
 data base implementation **GNU**
 sample **GNU**
 general considerations and restrictions, planning **HLPI**
 generation of **LLC**
 a DBD **DIAG, IRDU**
 a PSB **DIAG, IRDU**
 an ACT **DIAG, IRDU**
 DL/I control blocks **LLC**
 length of names **LLC**
 execution program (macro DLZNN) **LLC**
 format **LLC**
 sample **LLC**

initialization program for control data base
 (macro DLZNNICT) **LLC**
 modules required for link-edit **LLC**
 sample **LLC**
 get call functions **CALL**
 Get commands **GNU, HLPI**
 GET NEXT **GNU**
 get next (hold) - GN, GHN **DIAG**
 GET NEXT (see GN) **CALL**
 GET NEXT command **HLPI**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 GET NEXT IN PARENT command **GNU, HLPI**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 get next within parent (hold) - GNP, GHNP **DIAG**
 GET NEXT WITHIN PARENT (see GNP) **CALL**
 GET request **DIAG**
 GET UNIQUE **GNU**
 get unique (hold) - GU, GHU **DIAG**
 GET UNIQUE (see GU) **CALL**
 GET UNIQUE command **HLPI**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 getting started **IRDU**
 GHN (get hold next) **GNU**
 GHN call, detailed description **CALL**
 GHNP (get hold next within parent) **GNU**
 GHNP call, detailed description **CALL**
 GHU (get hold unique) **GNU**
 GHU call, detailed description **CALL**
 GN (get next) **GNU**
 GN call **CALL**
 definition of function **CALL**
 detailed description **CALL**
 for sequential retrieval **CALL**
 performance **CALL**
 programming hints **CALL**
 GN, GHN - get next (hold) **DIAG**
 GNP (get next within parent) **GNU**
 GNP call **CALL**
 definition of function **CALL**
 detailed description **CALL**
 example **CALL**
 GNP, GHNP - get next within parent (hold) **DIAG**
 GOBACK (COBOL) statement **CALL, DIAG, GNU**
 gozinto graph **LLC**
 application **LLC**
 concept **LLC**
 definition **LLC**
 grouping data elements with keys **ADBD**
 grouping segments **GNU**
 GSCD (get SCD) call **CALL, DIAG**
 format of in Assembler **DIAG**

format of in PL/I **DIAG**
 status codes **DIAG**
 use with COBOL **DIAG**
 GU (get unique) **GNU**
 GU call **CALL**
 definition of function **CALL**
 detailed description **CALL**
 for sequential processing **CALL**
 use of **CALL**
 GU, GHU - get unique (hold) **DIAG**

H

HANDLE ABEND command **HLPI**
 in programming example **HLPI**
 hardware requirements **ADBD**
 hashing method example **DBA**
 HD (hierarchical direct access method) **GNU**
 characteristics **GNU**
 data base in physical storage **GNU**
 data base reorganization **GNU**
 direct access pointers **GNU**
 inserts and deletes **GNU**
 HD access method **DBA, DIAG**
 HD characteristics **GNU**
 indexed processing **GNU**
 randomized access **GNU**
 HD data base record structures **DIAG**
 HD data bases **RDU**
 coding examples **RDU**
 primary indexed access **RDU**
 primary randomized access **RDU**
 secondary indexed access **RDU**
 control statements, format **RDU**
 ACCESS, for primary indexed **RDU**
 ACCESS, for primary randomized **RDU**
 ACCESS, for secondary indexed **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FIELD **RDU**
 FINISH **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 how to define **RDU**
 input structure and rules **RDU**
 HD data format **DIAG**
 HD indexed data base, loading of **GNU**
 HD indexed physical storage **DIAG**
 HD main data base **GNU**
 HD options **ADBD**
 HD parameters **DBA**
 adjusting options **DBA**
 HD primary index **DBA**
 data base **GNU**
 HD Randomized and HD Indexed, differences **DBA**
 HD randomized data base loading **GNU**
 HD randomized data base record **DIAG**
 HD randomized physical storage **DIAG**
 HD randomizing module errors **DIAG**
 HD reorganization **DBA**
 reload **DBA**
 rules and restrictions **DBA**
 reorganizing an HD data base **DBA**
 unload **DBA**

HD reorganization function, IUG **DIAG**
 HD reorganization reload **DBA**
 HD reorganization reload utility **DBA, DIAG, RDU, RRG**
 coding examples **RDU**
 control statements (REW) **IRDU, RDU**
 description **IRDU, RDU**
 doing a reload restart **IRDU, RDU**
 example of reload output statistics **RDU**
 illustration **IRDU**
 input/output **IRDU, RDU**
 job control statements **RDU**
 parameter statement **RDU**
 parameters **IRDU**
 HD reorganization unload backup **RRG**
 HD reorganization unload utility
 (DLZURGU0) **DBA, DIAG, GNU, RRG**
 coding example **RDU**
 control statements **IRDU, RDU**
 CHKPT **RDU**
 REW **RDU**
 description **IRDU, RDU**
 doing a selective unload **IRDU, RDU**
 coding example **RDU**
 parameter statement **RDU**
 example of unload output statistics **RDU**
 features of **RRG**
 illustration **IRDU**
 input **IRDU**
 input to **RRG**
 input/output **RDU**
 job control statements **RDU**
 output **IRDU**
 output from **RRG**
 parameter statement **RDU**
 parameters **IRDU**
 HD secondary index **DBA**
 HD single reorganization function, IUG **DIAG**
 HD space search algorithm **DBA**
 HD unload sequence checking **DIAG**
 HD, sequencing by field value for **DBA**
 HDAM **HLPI, LLC**
 direct processing **HLPI**
 presort for loading optional **HLPI**
 root segment sequence **HLPI**
 used for control data base **LLC**
 used for parts data base **LLC**
 with VARIABLE option **HLPI**
 HDAM data base record structures **DIAG**
 HDAM data base reorganization **DBA**
 HDAM data bases **RDU**
 coding examples **RDU**
 control statements, format **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FIELD **RDU**
 FINISH **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 how to define **RDU**
 input structure and rules **RDU**
 HDAM (hierarchical direct access method) **DIAG**
 data format **DIAG**
 physical storage **DIAG**
 randomizer **DBA**
 root segments returned in physical sequence **CALL**
 use of special characters **CALL**
 HDBFR **DIAG**
 operand

coding requirements **RDU**
 parameter statement **GNU**
 parameter **LLC, RDU**
 DLZACT macro **GNU**
 pool control **DBA**
 header specification **GNU**
 health care **GIM**
 health care application **ABBD**
 patient data base **ABBD**
 purchasing data base **ABBD**
 help panels
 description **IRDU**
 IMF, sample
 description **IRDU**
 interactive facilities **GNU**
 IUG, sample
 description **IRDU**
 sample **IRDU**
 using **IRDU**
HIDAM (hierarchical indexed direct
 access method) **DIAG, HLPI, LLC**
 data format **DIAG**
 index data base **CALL**
 physical storage **DIAG**
 presort for loading **HLPI**
 sequential or direct processing **HLPI**
 used for control data base **LLC**
 used for parts data base **LLC**
 with VARIABLE option **HLPI**
 with VSAM **LLC**
HIDAM data bases **RDU**
 coding example **RDU**
 control statements, format **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FIELD **RDU**
 FINISH **RDU**
 LCHILD **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 data base record structures **DIAG**
 data base reorganization **DBA**
 data base, primary index specification in the **DBA**
 how to define **RDU**
 (see also primary INDEX data bases) **RDU**
 input structure and rules **RDU**
HIDAM/HDAM secondary index **DBA**
 hierarchical data structure **GIM, GNU, HLPI**
 segments and their relationships **GIM**
 hierarchical direct
 access method (HDAM) **DIAG**
 data formats **DIAG**
 organization characteristics **DIAG**
 hierarchical direct access method (HD) **GNU**
 description of **GNU**
 prefix pointer **GNU**
 types of primary access **GNU**
 indexed access **GNU**
 randomized access **GNU**
 hierarchical direct organization **DIAG**
 hierarchical indexed direct access method (HIDAM) **DIAG**
 hierarchical indexed sequential access method (HISAM) **GNU**
 hierarchical levels **HLPI**
 maximum number of in data base record **HLPI**
 parent segment **HLPI**
 hierarchical levels, maximum **CALL**
 hierarchical path **HLPI, LLC**
 defined **LLC**
 following discontinued **LLC**
 processing terminated **LLC**
 hierarchical path and key feedback area **CALL**
 hierarchical sequence **GNU, HLPI**
 definition **ABBD**
 parent segments **HLPI**
 segments referenced in **HLPI**
 sequential organization **ABBD**
 hierarchical sequential access method (HSAM) **GNU**
 hierarchical sequential organization (HS) **DIAG**
 hierarchical structure **ABBD, CALL**
 compared to network **ABBD**
 definition **ABBD**
 hierarchical structure changes **DBA**
 combining segments **DBA**
 procedure **DBA**
 sequence of segment types **DBA**
 procedure **DBA**
 hierarchical structure, change of **CALL**
 hierarchical structures **DBA**
 hierarchy **DBA**
 data base **HLPI**
 example **DBA**
 illustrated **DBA**
 numbering sequence **DBA**
 path **DBA**
 sequence **DBA**
 high level language **DIAG**
 high level language debugging **CALL, DIAG, CALL**
 High Level Programming Interface (see HLPI) **HLPI**
 description **GNU**
 high usage modules **DBA**
 high usage paths **ABBD**
 high-level languages (see application programs, high-level
 languages) **LLC**
HISAM **ABBD, HLPI**
 disadvantages **ABBD**
 method of data storage **ABBD**
 presort for loading **HLPI**
 reorganizing a HISAM data base **DBA**
 sequential or direct processing **HLPI**
 uses **ABBD**
HISAM (hierarchical indexed sequential access method) **GNU**
 physical storage of a data base record **GNU**
 reorganization reload utility (DLZURRL0) **GNU**
 reorganization unload utility (DLZURUL0) **GNU**
HISAM and SHISAM indexes **DBA**
HISAM data base, reorganizing a **DBA**
HISAM data bases **RDU**
 coding example **RDU**
 control statements, format **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FIELD **RDU**
 FINISH **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 how to define **RDU**
 input structure and rules **RDU**
HISAM physical storage **DIAG**
HISAM reload procedures **RRG**
HISAM reorganization function, IUG **DIAG**
HISAM reorganization reload utility (DLZURRL0) **DBA, DIAG, RDU, RRG**
 coding example **RDU**
 control statement (L) **RDU**
 control statements **IRDU**
 description **IRDU, RDU**

- example of reload output statistics **RDU**
- illustration **IRDU**
- input/output **IRDU, RDU**
- job control statements **RDU**
- parameters **IRDU**
- HISAM reorganization unload **DBA**
 - reload **DBA**
 - unload **DBA**
- HISAM reorganization unload backup **RRG**
- HISAM reorganization unload utility (DLZURULO) **DIAG, RDU, RRG**
 - coding example **RDU**
 - control statement (R) **RDU**
 - control statements **IRDU**
 - description **IRDU, RDU**
 - example of unload output statistics **RDU**
 - illustration **IRDU**
 - input **IRDU**
 - input/output **RDU**
 - job control statements **RDU**
 - output **IRDU**
 - parameters **IRDU**
- HISAM Reorganization Unload/Reload Utilities **RRG**
 - Data Set Recovery utility **RRG**
 - HISAM Reload Procedures **RRG**
- HISAM unload output used as DUMPIN **DIAG**
- HLPI (High Level Programming Interface) **GNU, HLPI** and CICS/VS **GNU**
 - arguments **GNU**
 - commands **GNU**
 - and DL/I functions **GNU**
 - GET NEXT **GNU**
 - GET NEXT IN PARENT **GNU**
 - GET UNIQUE **GNU**
 - INSERT **GNU**
 - example **GNU**
 - functions **GNU**
 - syntax **DIAG**
 - description **GNU, HLPI**
 - elements of command language **GNU**
 - format of commands **GNU**
 - functions **GNU, HLPI**
 - introduction **HLPI**
 - languages written with **GNU**
 - options **GNU**
 - scheduling command **GNU**
 - syntax **GNU**
 - trigger, use of **GNU**
- HOLD calls **CALL**
- host language
 - restrictions **GNU**
 - use and features, restrictions **HLPI**
- HS access methods **DBA**
- HS organization characteristics **DIAG**
- HSAM (hierarchical sequential access method) **ADBD, GNU, HLPI**
 - access methods **DBA**
 - considerations **GNU**
 - key field not required **HLPI**
 - method of data storage **ADBD**
 - presort for loading **HLPI**
 - sequential processing **HLPI**
 - updating restriction **HLPI**
 - uses **ADBD**
- HSAM data bases **RDU, RRG**
 - coding example **RDU**
 - control statements, format **RDU**
 - DATASET **RDU**
 - DBD **RDU**
 - DBDGEN **RDU**
 - END **RDU**

- FIELD **RDU**
- FINISH **RDU**
- SEGM **RDU**
- control statements, summary **RDU**
- how to define **RDU**
- input structure and rules **RDU**
- updates to **CALL**
- HSAM or HISAM segment format **DIAG**
- HSAM physical storage **DIAG**
 - of a logical data structure **GNU**
- HSBFR **DIAG**
 - operand
 - coding requirements **RDU**
 - parameter statement **GNU**
 - parameter **RDU**
 - DLZACT macro **GNU**

I

- I/O area **CALL**
 - for Assembler **CALL**
 - for COBOL **CALL**
 - for PL/I **CALL**
 - for RPG II **CALL**
 - length of **CALL**
 - name of **CALL**
 - segment **GNU, HLPI**
- I/O for RPG II, FROM|INTO option **CALL**
- IBM System/360 Data Base Organizing and Maintenance Processor **LLC**
- IBM System/370 Chained File - DL/I Bridge (see Chained File - DL/I Bridge) **LLC**
- ID field in an FSE **DIAG**
- ID statement **RDU**
- identification, checkpoint **HLPI**
- identifier **ADBD**
 - definition **ADBD**
- identifier, command (see trigger) **HLPI**
- identifying modules in dumps **DIAG**
- image copy function, IUG recovery **DIAG**
- image copy utility **DIAG, GNU, RRG**
 - data set **GNU**
- IMF (Interactive Macro Facilities) **GNU**
 - ACTGEN **GNU**
 - data entry panels **IRDU**
 - as related to macros **IRDU**
 - DBDGEN **GNU**
 - description **IRDU**
 - functions **GNU, IRDU**
 - help panels **IRDU**
 - how it works **IRDU**
 - introduction **IRDU**
 - macros **IRDU**
 - menu panels **IRDU**
 - migrating from ELIAS **IRDU**
 - operating characteristics **IRDU**
 - PSBGEN **GNU**
 - using **GNU**
- IMF processes
 - ACT definition **IRDU**
 - ACT generation **IRDU**
 - DBD definition **IRDU**
 - DBD generation **IRDU**
 - ELIAS migration **IRDU**
 - introduction **IRDU**
 - PSB definition **IRDU**
 - PSB generation **IRDU**
- implementation

considerations **DBA**
 of data base **GNU**
 gross PERT chart **GNU**
 project approach **GNU**
 project cycle **GNU**
 sample project plan **GNU**
 overview, data base **GNU**
 phase
 first to implement **ABDD**
 requirements for success **ABDD**
 requirements
 data access requirements **ABDD**
 defining **ABDD**
 performance considerations **ABDD**
 recovery requirements **ABDD**
 security requirements **ABDD**
 structural considerations **ABDD**
 system **DBA**
 technique, logical relationships **GNU**
 implementing an application **ABDD**
 implementation **ABDD**
 first phase **ABDD**
 schedule **ABDD**
 implementation plan **ABDD**
 developing **ABDD**
 implementation in phases **ABDD**
 implicit definition **DBA**
 implied SSAs **CALL**
 improving data base performance **DBA**
 prefix resolution **DBA**
 prefix update **DBA**
 preorganization **DBA**
 protecting your data base **DBA**
 reorganization utilities **DBA**
 scan **DBA**
 tools for **DBA**
IMS/VS GNU
IMSCOMP operand coding requirements **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 inbound retrieve calls **DIAG**
INCLUDE statement **CALL**
 Assembler program **CALL**
 COBOL program **CALL**
 PL/I program **CALL**
 RPG II program **CALL**
 incompatibilities, DL/I and IMS **DBA**
INCORE DIAG
 tables, multiple **DIAG**
 independence **GNU**
 data **GNU**
 device **GNU**
 independent study program **DBA**
index GNU
 pointer segment **GNU**
 secondary indexes **GNU**
 secondary (see secondary index) **GNU**
 source segment **GNU**
 secondary indexes **GNU**
 target segment **GNU**
 secondary indexes **GNU**
 use of for HD data bases **GNU**
 index data base **ABDD, DBA**
 definition of **DBA**
 duplicate data fields **DBA**
 referencing the **DBA**
 retrieval from **CALL**
INDEX operand coding requirements **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 index pointer segments, where defined **RDU**
 in primary INDEX data bases **RDU**
 in secondary INDEX data bases **RDU**
 index root segments **DBA**
 additional data in **DBA**
 index sequencing **DBA**
 index source segment, how to define **RDU**
 index suppression **CALL**
 index target segment, how to define **RDU**
 indexed access **DBA**
 characteristics **ABDD, DBA**
 HD **GNU**
 overview of **GNU**
 how it works **ABDD**
 index data base **ABDD**
 pointers **ABDD**
 primary data base **ABDD**
 index data base **ABDD**
 method of data storage **ABDD**
 overview of **DBA**
 uses **ABDD**
 indexed data **CALL**
 indexed direct access **DBA**
 indexed processing **GNU**
 auxiliary storage **GNU**
 indexed processing, HD data bases **GNU**
indexes DBA
 defining **DBA**
 HD primary **DBA**
 HD secondary **DBA**
 HIDAM and HDAM **DBA**
 HISAM and SHISAM **DBA**
 for HIDAM and HDAM, defining **DBA**
 secondary **DBA, GNU**
INDEXTRC (X'70') DIAG
 indicator, RQDLI commands **GNU**
 information menu **IRDU**
 sample **IRDU**
 information, help
 using **IRDU**
 initial generation mode **LLC**
 description **LLC**
 details **LLC**
 prerequisites **LLC**
 processing **LLC**
 use **LLC**
 initial load function, IUG **DIAG**
 initial loading of a data base **CALL**
 initial value, defining an **DBA**
 initialization **GNU**
 aids
 batch **DIAG**
 MPS batch **DIAG**
 online **DIAG**
 and termination routines, DL/I **DBA**
 DL/I batch system **GNU**
 errors **HLPI**
 job control language **GNU**
 requirements **GNU**
 requirements, MPS **GNU**
 module, DL/I **GNU**
 MPS batch partition **GNU**
 of CICS/VS **DBA**

- of control data base **LLC**
- of the DL/I online system **GNU**
- insert **DBA, GNU**
 - command **GNU**
 - and positioning **GNU**
 - format of **GNU**
- insert - ISRT **DIAG**
- insert call path **DIAG**
- INSERT command **HLPI**
 - function **CALL, HLPI**
 - definition of **CALL**
 - for data base load **CALL**
 - in programming example **HLPI**
 - parentage **HLPI**
 - position pointer **HLPI**
 - segment selection **HLPI**
 - status codes **HLPI**
 - syntax **HLPI**
- insert rule **GNU**
 - logical child segment **GNU**
 - logical parent segment **GNU**
 - physical parent segment **GNU**
- insert rule for logical relationships, violations **CALL**
- insert rules **DBA, HLPI**
 - logical parent **DBA**
- insert, multiple segments **HLPI**
 - inserting **HLPI**
- inserting multiple segments into a data base **GNU**
- inserting segments **DBA, GNU**
 - field level sensitivity **GNU**
 - in an HD data base **GNU**
- INSERT command **GNU**
 - with a secondary index **GNU**
- insertion of root segment **CALL**
- insertion of segment **CALL**
 - with path call **CALL**
- insertion rule (see RULES operand) **GNU, RDU**
 - logical relationships **GNU**
- installation plan PERT chart **GNU**
- installation requirements **LLC**
 - DL/I bridge **LLC**
 - initial generation **LLC**
 - updating **LLC**
- installing additional applications **DBA**
 - data base considerations **DBA**
 - existing data bases modified **DBA**
 - existing data bases unchanged **DBA**
 - new data bases and existing data bases modified **DBA**
 - implementation considerations **DBA**
 - machine requirements **DBA**
 - storage requirements **DBA**
 - real **DBA**
 - virtual **DBA**
- installing DL/I dialogs **IRDU**
 - on CMS **IRDU**
- installing the DL/I documentation aid facility **RDU**
- installing the ISQL Extract Defines Utility **RDU**
- instructor schedules
 - combining local view **ADBD**
 - defining application programs **ADBD**
 - frequency of use **ADBD**
 - local view as input **ADBD**
 - online vs batch **ADBD**
 - requirement for **ADBD**
 - type of access **ADBD**
- instructor skills
 - combining local view **ADBD**
 - defining application programs **ADBD**
 - frequency of use **ADBD**
 - local view as input **ADBD**

- online vs batch **ADBD**
- requirement for **ADBD**
- integers 'ij' **GNU**
- integrity **GNU**
 - CICS/VS **GNU**
 - dynamic transaction backout **GNU**
 - file **GNU**
 - intent scheduling **GNU**
 - maintaining data **DBA**
 - program isolation **GNU**
- intent propagation, implications of **DBA**
- intent scheduling
 - and program isolation **DBA**
 - comparing with program isolation **GNU**
 - rules **DIAG**
- interactive facilities **DIAG, GNU**
 - data entry panels **GNU**
 - description **IRDU**
 - dialogue forms **GNU**
 - functions **DIAG**
 - getting started **IRDU**
 - help panels **GNU**
 - menu panels **GNU**
 - using **GNU**
- Interactive Macro Facilities (IMF) **DIAG, GNU, IRDU**
 - ACT module flow **DIAG**
 - components working with **DIAG**
 - DBD module flow **DIAG**
 - creating tables **DIAG**
 - phase grouping **DIAG**
 - save areas **DIAG**
 - ELIAS migration **DIAG**
 - error conditions
 - abnormal ends **DIAG**
 - error detecting **DIAG**
 - ISPF error log **DIAG**
 - modules **DIAG**
 - overview **DIAG**
 - panel information **DIAG**
 - PSB module flow **DIAG**
 - creating tables **DIAG**
 - phase grouping **DIAG**
 - save areas **DIAG**
 - table creation **DIAG**
 - ACT Tables **DIAG**
 - DBD tables **DIAG**
 - PSB tables **DIAG**
 - table summary **DIAG**
 - using **GNU**
- Interactive SQL (ISQL) **DBA, RDU**
- interactive system productivity facility (see ISPF) **GNU, IRDU**
- Interactive Utility Generation (IUG) **GNU, IRDU**
 - abnormal ends **DIAG**
 - components work with **DIAG**
 - data base label information **DIAG**
 - delete data option **DIAG**
 - detecting **DIAG**
 - display data option **DIAG**
 - error conditions **DIAG**
 - function, other IUG **DIAG**
 - ISPF error log **DIAG**
 - ISQL EXTRACT DEFINE **DIAG**
 - job streams **DIAG**
 - Backout utility **DIAG**
 - Change Accumulation utility **DIAG**
 - Extract Define utility **DIAG**
 - HD Reorganization Reload utility **DIAG**
 - HD Reorganization Unload utility **DIAG**
 - HISAM Reorganization Reload utility **DIAG**
 - HISAM Reorganization Unload utility **DIAG**

- Image Copy utility **DIAG**
- Initial Load utility **DIAG**
- Log Print utility **DIAG**
- Partial Reorganization, Part 1, utility **DIAG**
- Partial Reorganization, Part 2, utility **DIAG**
- Prefix Resolution utility **DIAG**
- Prefix Update utility **DIAG**
- Preorganization utility **DIAG**
- Recovery utility **DIAG**
- Scan utility **DIAG**
- Trace Print utility **DIAG**
- module **DIAG**
- overview **DIAG**
- panel information **DIAG**
- problem determination module flow **DIAG**
- recovery module flow **DIAG**
 - backout **DIAG**
 - change accumulation **DIAG**
 - image copy **DIAG**
 - log print **DIAG**
 - recovery **DIAG**
- reorganization/load module flow **DIAG**
 - HD reorganization **DIAG**
 - HD single reorganization **DIAG**
 - HISAM reorganization **DIAG**
 - initial load **DIAG**
 - partial reorganization **DIAG**
 - prefix resolution **DIAG**
 - prefix update **DIAG**
 - preorganization **DIAG**
 - scan **DIAG**
- resume interrupted job option **DIAG**
- tables **DIAG**
 - control **DIAG**
 - Default **DIAG**
 - part **DIAG**
 - sort **DIAG**
- using **GNU**
- interactively creating a DBD **GNU**
- interactively creating a PSB **GNU**
- interface (RPG II) with an application program **GNU**
- interface components **GNU**
 - RPG II **GNU**
- interface module **GNU**
- interface to DL/I, program structure **GNU**
- interface, DLZTRACE user exit **DIAG**
- interfaces
 - CICS/VS ISC to local DL/I **DIAG**
 - CICS/VS ISC to remote DL/I PRH **DIAG**
 - for remote data bases **DIAG**
 - local DL/I to CICS/VS ISC **DIAG**
 - mixed **HLPI**
 - Remote DL/I PRH to CICS/VS ISC **DIAG**
- interfacing with DL/I, ways of **GNU**
- internal error termination
 - CICS/VS **DBA**
 - DL/I **DBA**
- interpreting dumps **DIAG**
 - after an abend message **DIAG**
- intersecting attributes **ABDD**
 - definition **ABDD**
- intersection data **ABDD, DBA, GNU, HLPI**
 - definition **ABDD**
- intersystem communication (ISC) **DBA, DIAG, CALL**
 - interface module (DFHISP) **DIAG**
 - limitations **DIAG**
- interviews **ABDD**
 - conducting **ABDD**
 - attitude of interviewers **ABDD**
 - causes of hostile attitude **ABDD**
 - establishing a good relationship **ABDD**
 - getting detail **ABDD**
 - organizing **ABDD**
 - setting up a team **ABDD**
 - with lower levels **ABDD**
 - with management **ABDD**
- INTO option **GNU, HLPI**
 - path call **GNU**
- inventory data base **GNU**
 - DBD example phase 2 **GNU**
 - load PSB phase 3 **GNU**
 - logical DBD phase 2 **GNU**
 - logical PSB phase 2 **GNU**
 - physical DBD phase 3 **GNU**
 - PSB, logical **GNU**
 - sample application description **GNU**
- inventory item segment (STPIITM), description **GNU**
- inventory location segment (STSILOC), description **GNU**
- inventory sample data base description **GNU**
- inverse relationships between primary keys **ABDD**
- inverted structure **CALL**
 - definition **ABDD**
- invocation of LLC/CC in DL/I **LLC**
- invoking the Extract Defines routine **RDU**
- IPF format
 - converting to ISPF format **IRDU**
- IPK182 assembler message **DIAG**
- ISC (intersystem communication support) **DBA, DIAG**
 - interface module address **DIAG**
 - limitations for application programs **DIAG**
 - mirror transaction **DIAG**
- isolated attributes **ABDD**
 - definition **ABDD**
- isolating duplicate values **ABDD**
- isolating repeating data elements **ABDD**
- ISPF **GNU**
 - description **IRDU**
 - error log **DIAG**
 - format
 - converting from IPF format **IRDU**
 - using **IRDU**
- ISQL **RDU**
 - accessing data using **DBA**
 - DA run routines **RDU**
 - examples of output **RDU**
- ISQL EXTRACT DEFINES utility **GNU, RDU**
 - control statements **IRDU**
 - creating extract routine **RDU**
 - job control statements **RDU**
 - creating SQL/DS table **RDU**
 - job control statements **RDU**
 - description **IRDU**
 - displaying ISQL routine **RDU**
 - example **RDU**
 - displaying SQL/DS table data **RDU**
 - examples **RDU**
 - extracting DL/I data **RDU**
 - installing **RDU**
 - invoking SQL/DS routine **RDU**
 - job control statements **RDU**
 - job stream **RDU**
 - modifying **RDU**
 - retrieving **RDU**
 - listing **RDU**
 - loading data into SQL/DS tables **RDU**
 - parameter statements **IRDU, RDU**
 - processing **RDU**
 - retrieving data from DL/I data base **RDU**
- ISQL query routines **RDU**
- DBDACCESS **RDU**

DBDBASIC **RDU**
 DBDFIELD **RDU**
 DBDHIER **RDU**
 DBDLCHLD **RDU**
 DBDNAMES **RDU**
 DBDSEGM **RDU**
 DLISTATS **RDU**
 executing **RDU**
 for Documentation Aid facility **RDU**
 printed report **RDU**
 PSBFIELD **RDU**
 PSBNAMES **RDU**
 PSBPCB **RDU**
 PSBSEG **RDU**
 PSBSEGV **RDU**
 WHEREDBD **RDU**
 WHEREFLD **RDU**
 WHERESEG **RDU**
 ISRT (insert) **DIAG, GNU**
 (load type) request **DIAG**
 ISRT call **CALL**
 detailed description (add) **CALL**
 detailed description (load) **CALL**
 fully qualified **CALL**
 programming hints **CALL**
 IUG **GNU**
 data base **GNU**
 label information **GNU**
 load **GNU**
 recovery **GNU**
 reorganization **GNU**
 data entry panels **IRDU**
 delete or display data **GNU**
 description **IRDU**
 EXTRACT DEFINES **GNU**
 functions **GNU, IRDU**
 general routines **IRDU**
 generating DL/I utility job streams **IRDU**
 help panels **IRDU**
 how it works **IRDU**
 introduction **IRDU**
 menu panels **IRDU**
 operating characteristics **IRDU**
 primary menu **IRDU**
 problem determination **GNU**
 resume interrupted job **GNU**
 using **GNU**
 IUG processes **IRDU**
 data base label information **IRDU**
 delete or display data **IRDU**
 introduction **IRDU**
 ISQL EXTRACT DEFINES **IRDU**
 load **IRDU**
 problem determination **IRDU**
 recovery **IRDU**
 resume interrupted job **IRDU**

J

JCB address **GNU**
 JCB address field in PCB (see reserved field in PCB) **CALL**
 JCB trace (JCBTRACE) **DIAG**
 table **DIAG**
 JCBTRACE (JCB trace) **DIAG**
 JCL
 (See job control statements)
 JCL example **GNU**

MPS batch program execution **GNU**
 restarting an MPS restart program **GNU**
 JCL requirements **GNU**
 for DL/I initialization **GNU**
 for DL/I MPS initialization **GNU**
 JCT (journal control table) **DBA, RDU, RRG**
 JCT (journal control table), CICS/VS **GNU**
 JDUMP **DIAG**
 job control **GNU**
 DBDGEN execution **GNU**
 execution of PSBGEN **GNU**
 for creating the online nucleus **GNU**
 online **GNU**
 COBOL **GNU**
 PL/I **GNU**
 RPG II **GNU**
 job control and logging **RRG**
 job control errors **DIAG**
 Job Control for DL/I Utilities
 Data Base Backout **RRG**
 Data Base Change Accumulation **RRG**
 Data Base Prefix Resolution **RRG**
 Data Base Prefix Update **RRG**
 Data Base Preorganization **RRG**
 Data Base Scan **RRG**
 Data Set Image Copy **RRG**
 Data Set Recovery **RRG**
 HD Reorganization Reload **RRG**
 HD Reorganization Unload **RRG**
 HISAM Reorganization Reload **RRG**
 HISAM Reorganization Unload **RRG**
 Log Print **RRG**
 Scan utility **RRG**
 job control parameters for DL/I **DIAG**
 job control statements **CALL, GNU, HLPI, RDU**
 and parameters **DIAG**
 batch **DIAG, HLPI**
 examples of **GNU**
 for ACTGEN **DIAG**
 for application program **CALL**
 for deleting a DMB **DIAG**
 for DLZDLTXX **DIAG**
 for logging **GNU**
 for MPS batch program execution **CALL**
 for MPS batch program using MPS restart **CALL**
 for trace print utility **DIAG**
 ISQL Extract Defines Utility **RDU**
 MPS batch **HLPI**
 MPS batch using MPS restart facility **HLPI**
 restarting MPS batch program **HLPI**
 trace print utility examples **DIAG**
 JOB names, entering **IRDU**
 job stream example
 ACT **IRDU**
 DBD **DIAG, IRDU**
 PSB **IRDU**
 job stream generation
 resuming **IRDU**
 job stream output **IRDU**
 saving **IRDU**
 journal buffer size (JCT) **GNU**
 maximum **GNU**
 minimum **GNU**
 journal control table (JCT), CICS/VS **DBA, GNU, RDU**
 example **RRG**
 journal, CICS/VS **GNU, RRG**

K

key
 concatenated key **ABDD**
 definition **ABDD**
 controlling **ABDD**
 definition **ABDD**
 grouping data elements with **ABDD**
 identifying **ABDD**
 concatenated **ABDD**
 primary **ABDD**
 secondary **ABDD**
 primary key **ABDD**
 avoiding many-to-many relationships between **ABDD**
 candidate **ABDD**
 definition **ABDD**
 identifying **ABDD**
 root key **ABDD**
 secondary key **ABDD**
 definition **ABDD**
 identifying **ABDD**
 key feedback **GNU, HLPI**
 DIBKFBL **GNU**
 retrieving **HLPI**
 key feedback area
 field level sensitivity **CALL**
 in PCB mask **CALL**
 length of **GNU**
 key feedback length **HLPI**
 truncation **HLPI**
 key field **CALL, DBA, CALL, GNU, HLPI**
 change of **CALL**
 definition **ABDD, HLPI**
 dependent segments **CALL**
 effect on insertion **HLPI**
 HDAM root segments not sequenced by **HLPI**
 root segment **CALL**
 segment qualification on **HLPI**
 key field in a data base record **GNU**
 key length **GNU**
 DIBKFBL **GNU**
 key of a data base record **GNU**
 key sequence **HLPI**
 key sequenced data set (KSDS) **GNU**
 key, concatenated **GNU, HLPI**
 KEYFDBK **DIAG**
 KEYFEEDBACK **HLPI**
 KEYLEN operand
 coding requirements **RDU**
 PCB statement **GNU**
 keyword restriction **GNU**
 keywords, reserved **HLPI**
 KSDS (key sequenced data set) **GNU**

in PSBGEN statement **RDU**
 PSBGEN statement **GNU**
 LANG= parameter (DLZACT) **DIAG**
 language interface module (DLZLI000) **CALL, DIAG**
 for Assembler program **CALL**
 for COBOL program **CALL**
 for PL/I program **CALL**
 for RPG II program **CALL**
 language interface program **DBA**
 languages **CALL**
 mixing **CALL**
 LAST option **HLPI**
 LCF (logical child first pointer) **GNU**
 LCHILD
 specification of logical child **DBA**
 specification of virtual logical child **DBA**
 statement **DBA**
 statement for the logical parent **DBA**
 LCHILD statement **GNU, RDU**
 format **RDU**
 for index target segments in HDAM and HIDAM data **RDU**
 for logical child segments in bidirectional logical
 relationships **RDU**
 for logical child segments in unidirectional logical
 relationships **RDU**
 for primary INDEX data bases **RDU**
 for primary index relationships in HIDAM data bases **RDU**
 for secondary INDEX data bases **RDU**
 NAME operand **GNU**
 PAIR operand **GNU**
 POINTER operand **GNU**
 RULES operand **GNU**
 LCL (logical child last pointer) **GNU**
 left-to-right ordering of children, defining **DBA**
 left-to-right ordering of segment types **DBA**
 length **HLPI**
 maximum field **HLPI**
 maximum segment **HLPI**
 length calculation, variable length concatenated segment **HLPI**
 length field **HLPI**
 definition **HLPI**
 in SEGLENGTH for variable length segments **HLPI**
 in variable length concatenated segment **HLPI**
 of variable length segments **CALL**
 length of I/O area **CALL**
 length of key feedback area **GNU**
 level **DBA**
 definition **ABDD, DBA**
 illustrated **DBA**
 levels, maximum number **CALL**
 in data base record **HLPI**
 LI statement **RDU**
 LIFO list
 ACT **DIAG**
 PSB **DIAG**
 limitation **LLC**
 insertion of segments not performed **LLC**
 updating of deletions not supported **LLC**
 limited data sharing **DBA**
 link
 double arrow **ABDD**
 example **ABDD**
 isolated attributes **ABDD**
 secondary key candidates **ABDD**
 single arrow **ABDD**
 between keys in color **ABDD**
 with concatenated keys **ABDD**
 link editing and compilation, online **GNU**
 link-editing **HLPI**

L

L command code **GNU**
 label information, entering
 description **IRDU**
 labels **GNU, HLPI**
 CICS/VS generated **GNU**
 in DIB **HLPI**
 reserved **GNU, HLPI**
 restriction **GNU**
 LANG operand coding requirements **RDU**
 coding requirements **RDU**
 in DLZACT TYPE=RPSB statement **RDU**

link-editing of application programs **LLC**
 linkage editing **CALL**
 Assembler program **CALL**
 COBOL program **CALL**
 PL/I program **CALL**
 RPG II program **CALL**
 linkage editor warnings **DIAG**
 ACT **DIAG**
 DLZDBH00 **DIAG**
 DLZNUCXX **DIAG**
 DLZODP **DIAG**
 online nucleus **DIAG**
LISTCAT
 checking catalog entry for data base **DIAG**
 verifying open/close information **DIAG**
LL field **HLPI**
LNAME operand coding requirements **RDU**
LO statement **RDU**
LOAD **GNU**
 command **GNU**, **HLPI**
 format of **GNU**
 function **HLPI**
 in programming example **HLPI**
 loading data bases **GNU**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 description **IRDU**
 load batch application program **HLPI**
 load call path **DIAG**
 load module, online nucleus **GNU**
 load of data base **CALL**
 ISRT call **CALL**
LOAD operand coding requirements **RDU**
LOAD processing **GNU**
 data base **GNU**
 load program **DBA**
 (see loading) **HLPI**
 basic loading **DBA**
 comparison of HD access methods **DBA**
 illustrated **DBA**
 processing option (**PROCOPT**) **DBA**
 HD randomizing and **HD** indexed differences **DBA**
 loading **HD** indexed or **HIDAM** data bases **DBA**
 loading **HD** randomized or **HDAM** data bases **DBA**
 prefix resolution **DBA**
 job control statements **DBA**
 utility programs **DBA**
 with logical relationships **DBA**
 with secondary indexes **DBA**
 samples **DBA**, **GNU**
 status code testing **DBA**
 load **PSB**, customer data base **GNU**
LOAD sample program **HLPI**
 loading **GNU**, **HLPI**
 a basic data base **GNU**
 a data base **DIAG**, **GNU**
 description **IRDU**
 and **HD** indexed data base **GNU**
 and **HD** randomized data base **GNU**
 basic **DBA**
 data bases **GNU**
 data bases with logical relationships and/or
 secondary indexes **GNU**
 HD indexed or **HIDAM** data bases **DBA**
 HD randomized or **HDAM** data bases **DBA**
 of two data bases with logical relationships and
 secondary indexes **GNU**
 process **GNU**
 requiring prefix resolution **DBA**
 utility programs **GNU**
 workfile **GNU**
 loading data into **SQL/DS** tables **RDU**
 loading programs, data base **DBA**
 local view **DBA**
 building **ABDD**
 analyzing associations **ABDD**
 considerations that might alter **ABDD**
 examples **ABDD**
 identifying keys **ABDD**
 logical relationship candidates **ABDD**
 secondary indexing candidates **ABDD**
 using the three step process **ABDD**
 considerations that might alter **ABDD**
 addition **ABDD**
 deletion **ABDD**
 replacement **ABDD**
 retrieval **ABDD**
 definition **ABDD**
 examples **ABDD**
 class schedules **ABDD**
 instructor schedules **ABDD**
 instructor skills **ABDD**
 identifying alternate processing sequences **ABDD**
 identifying intersecting attributes **ABDD**
 identifying relationships (mapping) **ABDD**
 definition **ABDD**
 logical relationship candidates **ABDD**
 mapping **ABDD**
 definition **ABDD**
 many-to-many **ABDD**
 one-to-many **ABDD**
 one-to-one **ABDD**
 secondary indexing candidates **ABDD**
 locating **ACBs** **DIAG**
 locating buffers **DIAG**
 locating **DMBs** **DIAG**
 locating **RPLs** **DIAG**
 locating the buffer pool **DIAG**
LOCKED option **HLPI**
 lockout function **HLPI**
LOG **DIAG**
 log file **CALL**
 specification in parameter statement **CALL**
 log medium **DBA**
 choosing **GNU**
 in batch **DBA**
 in **MPS** batch **DBA**
 in online **DBA**
LOG operand, parameter statement **GNU**
LOG parameter **RDU**, **RRG**
 log print function, **IUG** problem determination **DIAG**
 log print utility (**DLZLOGP0**) **DBA**, **DIAG**, **RDU**, **RRG**
 coding examples **RDU**
 control statements **IRDU**, **RDU**
 LI **RDU**
 LO **RDU**
 LS **RDU**
 description **IRDU**, **RDU**
 illustration **IRDU**
 input **IRDU**
 input to **RRG**
 input/output **RDU**
 job control statements **RDU**
 keyword record format to **DSECT** mapping **RDU**
 output **IRDU**
 output from **RRG**

parameters **IRDU**
 sample record contents in keyword format **RDU**
 types of log files **RRG**
 log print utility, data base **GNU**
 log print, IUG recovery **DIAG**
 log
 data base **GNU**
 DL/I **RRG**
 logging **DBA, RRG**
 and CICS/VS journal **RRG**
 and performance **DBA**
 asynchronous option **DBA**
 choosing log medium **DBA**
 CICS/VS environment **DBA**
 MPS batch environment **DBA**
 normal batch environment **DBA**
 CICS/VS system journal file **RRG**
 definition of **RRG**
 disk **DBA**
 disk logging **DBA**
 in batch **DBA**
 in online **DBA**
 DL/I log **RRG**
 facility, DL/I **DBA**
 for modified data **RRG**
 forward recovery log from backout utility **RRG**
 log write times **DBA**
 multifile system log tapes **DBA**
 multivolume system log tapes **DBA**
 performance considerations for **DBA**
 logging and job control **RRG**
 and job control **RRG**
 and performance **GNU**
 and the UPSI byte **RRG**
 facility **GNU**
 job control statements for **GNU**
 logic qualification, Boolean **CALL**
 logical access path **GNU**
 logical child **DBA, GNU**
 accessing in a physical DBD **GNU**
 coding **GNU**
 counter **DBA**
 defining a normal **DBA**
 defining the segment **DBA**
 delete rules (bidirectional relationships only) **DBA**
 deletion rules **GNU**
 first pointer (LCF) **GNU**
 last pointer (LCL) **DBA, GNU**
 LCHILD specification of **DBA**
 parent operand for **DBA**
 placement of in bidirectional relationships **DBA**
 segment **GNU**
 format **GNU**
 defining the **DBA**
 type **GNU**
 defining the **DBA**
 virtual logical child specification of **DBA, GNU**
 logical child first **DBA**
 first **DBA**
 last **DBA**
 logical child segment **CALL, HLPI**
 definition **HLPI, RDU**
 insertion of **CALL**
 LCHILD specification of **DBA**
 loading **HLPI**
 loading of **CALL**
 multiple segment insertion restriction **HLPI**
 physical delete rule **CALL**
 logical children

physical structure rules **DBA**
 logical data bases **DBA, GNU, HLPI, RDU**
 after relating CUSTOMER and INVENTORY data bases **GNU**
 allowable structures **DBA**
 coding example **RDU**
 control statements, format **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FINISH **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 defining segments for **DBA**
 base segments **DBA**
 segment name **DBA**
 defining structures for **DBA**
 left-to-right ordering of children **DBA**
 parent/child relationships **DBA**
 definition **ABDB, DBA**
 fields for **DBA**
 how to define **RDU**
 (see also defining logical relationships in HD,
 HDAM, and HIDAM) **RDU**
 input structure and rules **RDU**
 logical data structure **HLPI**
 logical DBD **GNU**
 accessing segments in **GNU**
 coding **GNU**
 logical parent **DBA, GNU**
 concatenated key (LPCK) **GNU**
 defining the sequence of **DBA**
 deletion rules **GNU**
 LCHILD statement **DBA**
 LCHILD statement for **DBA**
 physical delete rule **CALL**
 pointer (LP) **GNU**
 segments **GNU**
 how to define **RDU**
 type **GNU**
 logical path **DBA**
 logical pointer summary for HD data bases **DIAG**
 logical record length **DBA**
 logical relations, special considerations **DBA**
 logical relationship resolution utilities **RDU**
 restrictions **RDU**
 logical relationship utilities
 data base prefix resolution **DIAG**
 data base prefix update **DIAG**
 data base scan **DIAG**
 workfile generator **DIAG**
 logical relationships **GNU**
 adding a **DBA**
 ACCESS operand, logical DBD statement **GNU**
 accessing through different path **ABDB**
 as direct access considerations **ABDB**
 as structural considerations **ABDB**
 bidirectional **ABDB, GNU**
 building logical relationships **GNU**
 BYTES operand of SEGM statement for a logical child **GNU**
 candidates **ABDB**
 coding
 a logical child **GNU**
 a logical DBD **GNU**
 in a physical DBD **GNU**
 concatenated segments **GNU**
 DATASET statement, logical DBD **GNU**
 DBD examples, physical **GNU**
 definition **ABDB**
 examples, illustrated **DBA**

HD support **GNU**
 inverting parent/child relationship **ADBD**
 loading data bases with **GNU**
 logical structure **ADBD**
 processing with **GNU**
 redundant data **ADBD**
 reorganizing a data base **DBA**
 steps in **DBA**
 replacement rules **GNU**
 unidirectional **ADBD**
 uses **ADBD**
 of utilities when adding **DBA**
 in health care application **ADBD**
 of pointers **ADBD**
 logical relationships, defining in HD, HDAM, and
 HIDAM data bases **RDU**
 coding example **RDU**
 control statements, summary **RDU**
 how to define **RDU**
 (see also LOGICAL data bases) **RDU**
 segment types used **RDU**
 logical child segments **RDU**
 logical parent segments **RDU**
 physical parent segments **RDU**
 virtual logical child segments **RDU**
 logical retrieve **DIAG**
 debugging aids **DIAG**
 internal macros **DIAG**
 DLZRCLL macro **DIAG**
 DLZREXT macro **DIAG**
 DLZRHDR macro **DIAG**
 DLZRTL macro **DIAG**
 linkage macros
 DLZRLNK C macro **DIAG**
 DLZRLNK D macro **DIAG**
 DLZRLNK M macro **DIAG**
 logical structure **ADBD**
 definition **ADBD**
 logical twin backward **DBA**
 pointer (LTB) **DBA, GNU**
 logical twin forward pointer (LTF) **GNU**
 logical twin pointers, suppression of **DBA**
 logical unit of work (LUW)
 and DL/I **RRG**
 defining **RRG**
 defining smaller LUWs **RRG**
 example of **RRG**
 maximum size for **RRG**
 recovery from **RRG**
 logical versus physical Relationships **DBA**
 logically related data bases **RRG**
 Data Base Prefix Resolution utility **RRG**
 Data Base Prefix Update utility **RRG**
 Data Base Preorganization utility **RRG**
 Data Base Scan utility **RRG**
 HD Reorganization Reload utility **RRG**
 steps to restore a data base **RRG**
 utilities for **RRG**
 logically related segments, processing **GNU**
 logs, need for **RRG**
 loop **LLC**
 description **LLC**
 detected **LLC**
 low usage module **DBA**
 low-level coding **LLC**
 numeric values in reverse order **LLC**
 usage **LLC**
 problems outside manufacturing **LLC**
 production planning **LLC**
 LP (logical parent pointer) **GNU**

LPCK (logical parent concatenated key) **GNU**
 LS statement **RDU**
 LTB (logical twin backward pointer) **GNU**
 LTF (logical twin forward pointer) **GNU**
 LUW (logical unit of work) **RRG**

M

machine requirements **DBA**
 minimum configuration **DBA**
 application profile **DBA**
 CICS/DOS/VS in VM/370 virtual machine **DBA**
 examples **DBA**
 units permitted **DBA**
 VSE **DBA**
 real storage requirements **DBA**
 application profile **DBA**
 example **DBA**
 macros **LLC**
 creating in IMF **IRDU**
 DLZNN **LLC**
 format **LLC**
 sample **LLC**
 DLZNNICT **LLC**
 sample **LLC**
 relating to data entry panels **IRDU**
 main processing sequence **CALL**
 main routine flow, MPS **DIAG**
 maintaining data integrity **DBA**
 management, data base user responsibility **GNU**
 manufacturing industry **GIM**
 many-to-many mapping
 procedure for avoiding **ADBD**
 many-to-many relationships
 avoiding between primary keys **ADBD**
 procedure **ADBD**
 between data aggregates **ADBD**
 between data elements **ADBD**
 between primary keys **ADBD**
 not allowed in hierarchy **ADBD**
 many-to-one relationships
 between data elements **ADBD**
 mapping
 between keys **ADBD**
 definition **ADBD**
 many-to-many **ADBD**
 procedure for avoiding **ADBD**
 master catalog, VSAM **GNU, LLC**
 master partition controller partition table (MPCPT) **DIAG**
 maximum
 field length **HLPI**
 number of hierarchical levels in data base record **HLPI**
 number of parent segments **GNU**
 number of segment types **HLPI**
 segment lengths **GNU, HLPI**
 MAXTASK operand coding requirements **RDU**
 MAXTASK parameter **GNU**
 DLZACT macro **GNU**
 MCT (monitoring control table) **RDU**
 MDLI - online test program **DIAG**
 menu panels
 IMF primary menu
 description **IRDU**
 interactive facilities **GNU**
 IUG load menu
 description **IRDU**
 sample **IRDU**

- IUG primary menu **IRDU**
- description **IRDU**
- IUG problem determination
- sample **IRDU**
- IUG recovery menu **IRDU**
- sample **IRDU**
- IUG reorganization menu
- description **IRDU**
- sample **IRDU**
- menu, information
- sample **IRDU**
- using **IRDU**
- messages **GNU, LLC**
- abend message summary **DIAG**
- diagnostic aids for **DIAG**
- DLZ001I **DIAG, GNU**
- DLZ002I **DIAG, GNU**
- DLZ020I **GNU**
- DLZ025I **DIAG**
- DLZ031I **GNU**
- DLZ068I **GNU**
- DLZ070I **GNU**
- DLZ082I **DIAG**
- DLZ083I **DIAG**
- DLZ084I **DIAG**
- DLZ085I **DIAG**
- DLZ089I **DIAG**
- DLZ090I **DIAG**
- DLZ095I **DIAG**
- DLZ096I **DIAG, GNU**
- DLZ097I **DIAG**
- DLZ100I **DIAG**
- DLZ102I **DIAG**
- DLZ105I **GNU**
- DLZ106I **GNU**
- DLZ108I **GNU**
- DLZ113I **GNU**
- DLZ260I **DIAG**
- DLZ261I **DIAG**
- DLZ262I **DIAG**
- DLZ263I **DIAG**
- DLZ264I **DIAG**
- DLZ265I **DIAG**
- DLZ266I **DIAG**
- DLZ267I **DIAG**
- DLZ268I **DIAG**
- DLZ380I **DIAG**
- DLZ400I **DIAG**
- DLZ476I **DIAG**
- DLZ772I **DIAG**
- DLZ796I **DIAG**
- DLZ797I **DIAG**
- DLZ798I **DIAG**
- DLZ799I **DIAG**
- DLZ800I **DIAG**
- DLZ801I **DIAG**
- DLZ802I **DIAG**
- DLZ803I **DIAG**
- DLZ804I **DIAG**
- DLZ806I **DIAG**
- DLZ807I **DIAG**
- DLZ808I **DIAG**
- DLZ830I **DIAG**
- DLZ831I **DIAG**
- DLZ832I **DIAG**
- DLZ841I **DIAG**
- DLZ844I **DIAG**
- DLZ845I **DIAG**
- DLZ847I **DIAG**
- DLZ848I **DIAG**

- DLZ850I **DIAG**
- DLZ855I **DIAG**
- DLZ860I **DIAG**
- DLZ861I **DIAG**
- DLZ862I **DIAG**
- DLZ863I **DIAG**
- DLZ864I **DIAG**
- DLZ868I **DIAG**
- DLZ938I **DIAG**
- DLZ942I **DIAG**
- DLZ943I **DIAG**
- IPK182 **DIAG**
- error **IRDU**
- method of distribution, LLC/CC in DL/I **LLC**
- migrating indexes
- using ELIAS **IRDU**
- minimum configuration **DBA**
- minimum DL/I configuration **GIM**
- mirror program (DFHMIR) **DIAG**
- mirror transaction **DIAG**
- mixed interfaces **HLPI**
- mixed use of interfaces, restriction **GNU**
- mnemonic status codes **DIAG**
- modifying data (see REPLACE command) **HLPI**
- modifying the DL/I documentation aid job streams **RDU**
- modifying the ISQL Extract Defines job stream **RDU**
- MODTRACE (X'10') **DIAG**
- module assembly **DIAG**
- module change level, identifying in dumps **DIAG**
- module flow
- ACT **DIAG**
- DBD **DIAG**
- ELIAS table migration **DIAG**
- PSB **DIAG**
- table migration **DIAG**
- module identification in dumps **DIAG**
- module name, identifying in dumps **DIAG**
- MODULE operand coding requirements **RDU**
- modules, IMF **DIAG**
- modulo or division method example **DBA**
- monitored DL/I events **DBA**
- monitoring
- control table (see MCT) **DBA, RDU**
- facilities (CMF) hooks, CICS/VS **DBA**
- facilities, activating the CICS/VS **DBA**
- monitoring tools **DBA**
- run and buffer statistics **DBA**
- buffer pool configuration **DBA**
- buffer statistics **DBA**
- examples of use **DBA**
- run statistics **DBA**
- MPCPT (master partition controller partition table) **DIAG**
- MPS (multiple partition support) **CALL, GNU, RDU**
- abnormal termination **GNU**
- and concurrent online execution **DBA**
- illustrated **DBA**
- and online considerations **GNU**
- batch and online differences **GNU**
- batch program execution **CALL**
- job control example **CALL**
- batch program using MPS restart **CALL**
- job control example **CALL**
- batch programs, executing with MPS **GNU**
- CICS/VS-DL/I interface **GNU**
- considerations **CALL**
- for RPG II **GNU**
- control blocks **DIAG**
- diagnostic aids for **DIAG**
- differences between batch, MPS, and online **GNU**

dynamically selecting MPS or non-MPS **RDU**
executing batch MPS programs **GNU**
executing batch program
example **RDU**
using restart facility, example **RDU**
executing CICS/VS with DL/I MPS **GNU**
job control statements **RDU**
main routine flow **DIAG**
notification of wait for a resource **CALL**
online restrictions **GNU**
parameter information **RDU**
programming considerations **GNU**
requirements **RDU**
restart facility **CALL, GNU, RDU**
description **CALL**
function code **RDU**
program execution example **CALL**
program restart, example **CALL**
restarting batch program, example **RDU**
restrictions for RPG II **GNU**
storage considerations **GNU**
transactions, defining (PCT) **GNU**
UPSI byte settings **RDU**
XECB
usage **DIAG**
waitlist dependence **DIAG**

MPS batch GNU, HLPI
compilation and link-editing **HLPI**
considerations and restrictions **HLPI**
execution **HLPI**
job control statements **HLPI**
program entry **HLPI**
program termination **HLPI**
reinitialization **GNU, HLPI**
during MPS restart **HLPI**
restart facility **HLPI**
considerations **HLPI**
description **HLPI**
job control statements **HLPI**

MPS batch application program CALL
job control example **CALL**
using MPS restart facility **CALL**
job control example **CALL**

MPS batch
control flow **DIAG**
initialization aids **DIAG**
partition initialization **GNU**
program execution **DBA**
program request handler **DIAG**
programs **GNU**
executing **GNU**
restart **RRG**
JCL example **RRG**
VSE checkpoint ID **RRG**
termination **DIAG**
user error aids **DIAG**

MPS data areas
batch communications area **DIAG**
batch wait states **DIAG**
master partition controller partition table **DIAG**

MPS initialization DIAG

MPS operation DBA
batch partition **DBA**
in MPS batch
data base system flow **DBA**
mirror task **DBA**
starting MPS **DBA**
system control flow **DBA**
system execution **DBA**
using restart facility **DBA**

online partition **DBA**
performance considerations **DBA**
restart facility **DBA**
description **DBA**
function code **DBA**
requirements **DBA**
system requirements **DBA**
types of program excluded from **DBA**
with CICS/VS **DBA**

MPS programs
with program isolation, storage for **CALL**
with VSE CHKPT **CALL**

MPS restart facility DIAG, GNU, HLPI
checkpoint
notification **DBA**
verification **DBA**
combined checkpoint verification **DBA**
considerations **DBA, HLPI**
description of **DBA, HLPI**
MPS reinitialization **DBA**
purging TSQ **DBA**
reinitializing the batch environment **DBA**
user restart programs **DBA**
using **GNU**

MPS restart program GNU
MPS system storage, DL/I DBA
multifile system log tapes **DBA**
multiple incore tables **DIAG**
multiple occurrences of data aggregate **ADBD**
choosing **ADBD**
multiple partition support (see MPS) **GNU, RDU**
multiple PCBs, position **GNU**
multiple positioning **CALL, HLPI**
example of **CALL**
specifying **DBA**
SSAs for **CALL**
use of **CALL**

multiple segment insertion **GNU, HLPI**
multiple values of data element **ADBD**
multivolume file dump/copy/restore **RRG**
multivolume system log tapes **DBA**
MXT parameter, CICS/VS **GNU**

N

N command code **GNU**
NAME operand **GNU**
DBD statement **GNU**
FIELD statement **GNU**
logical **DBD** statement **GNU**
of **SEGM** statement for a logical child **GNU**
SEGM statement **GNU**
SEGM statement for a logical child **GNU**
SEGM statement for a logical parent **GNU**
SEGM statement in logical **DBD GNU**
SENSG statement (**PSBGEN**) **GNU**
VIRFLD statement (**PSBGEN**) **GNU**
NAME operand coding requirements **RDU**
in **DBD** statement **RDU**
HD data bases **RDU**
HDAM data bases **RDU**
HIDAM data bases **RDU**
HISAM data bases **RDU**
HSAM data bases **RDU**
LOGICAL data bases **RDU**
primary **INDEX** data bases **RDU**
secondary **INDEX** data bases **RDU**
SHISAM data bases **RDU**

SHSAM data bases **RDU**
 in FIELD statement **RDU**
 /CK system related fields **RDU**
 /SX system related fields **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 in INITIAL type DLZSLC statement **RDU**
 in LCHILD statement **RDU**
 index target segments in HDAM and HIDAM data bases **RDU**
 logical child segments in HD, HDAM, and HIDAM
 data bases **RDU**
 primary INDEX data bases **RDU**
 primary index relationships in HIDAM data bases **RDU**
 secondary INDEX data bases **RDU**
 in SEGM statement **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 logical child segments **RDU**
 LOGICAL data bases **RDU**
 primary INDEX data bases **RDU**
 secondary INDEX data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 virtual logical child segments **RDU**
 in SENFLD statement **RDU**
 in SENSEG statement **RDU**
 in VIRFLD statement **RDU**
 in XDFLD statement **RDU**
 name, syntax **HLPI**
 naming conventions **DBA**
 additional for DL/I **DBA**
DBD DBA
 field **DBA**
 file-id **DBA**
 filename **DBA**
 for sample application **GNU**
 introduction **ABBD**
 job name **DBA**
 phase name **DBA**
 program name **DBA**
PSB DBA
 recommended **ABBD, DBA**
 additional for DL/I **ABBD, DBA**
 segment **DBA**
 used in sample application **ABBD**
 network structure **ABBD**
 compared to hierarchical **ABBD**
 definition **ABBD**
 networks **LLC**
NEXT MACRO entries
 description **IRDU**
 example **IRDU**
NLT (nucleus load table) **DBA**
 nodes **LLC**
 non-HD, sequencing by field value for **DBA**
 non-key fields
 qualification on **GNU**
 segment qualification on **HLPI**
 nonconcatenated segments (see **SOURCE** operand coding

requirements for LOGICAL data bases) **RDU**
NOREWIND option **RRG**
 nucleus **GNU**
 assembly listing for online generation **GNU**
 control statement listing for online **GNU**
 diagnostics for online generation **GNU**
 job control for creating the online **GNU**
 load table (NLT) **DBA**
 online load module **GNU**
NULLVAL operand coding requirements **RDU**
 number **HLPI**
 of hierarchical levels in data base record, maximum **HLPI**
 of segment types, maximum **HLPI**

O

object segment **GNU, HLPI**
 example of **GNU**
 obtaining a PSB **HLPI**
 obtaining the address of the PCB **GNU**
 occurrence, definition **GNU**
 off-line recovery **DBA**
OFFSET option **HLPI**
 one-to-many relationships
 between data aggregates **ABBD**
 between data elements **ABBD**
 one-to-one relationships
 between data aggregates **ABBD**
 between data elements **ABBD**
 online **DBA, HLPI**
 abnormal termination of **CALL**
 application program **CALL**
 applications **GNU**
 DL/I relationship **GNU**
 program interface **DIAG**
 Assembler working storage **CALL**
CICS/VS trace tables **CALL**
 compilation and link-editing **HLPI**
 considerations and restrictions **HLPI**
 execution **HLPI**
 issuing DL/I call **CALL**
 nucleus **DBA**
 operation **DBA**
PCB scheduling call **CALL**
 system environment **DBA**
 termination call **CALL**
 termination of task **HLPI**
 test program **DBA**
 online ASRAabend debugging **DIAG**
 online call processing **DIAG**
 online data base contention, minimizing **GNU**
 online data base processing **DBA**
 online data base system **GNU**
 online environment **CALL, DBA, CALL, GNU**
 online environment tracing (DLZTRACE)
 (See also online trace facilities)
 debugging **DIAG**
 defining the extrapartition data set **DIAG**
 example **DIAG**
 general trace **DIAG**
 opening the CICS/VS device **DIAG**
 trace output **DIAG**
 tracing control **DIAG**
 online exit conditions **DIAG**
 online formatted system dump **DIAG**
 online formatted task dump **DIAG**
 online initialization **DIAG**
 aids **DIAG**

with CICS/VS **DIAG**
 online interface **DIAG**
 online nucleus **GNU**
 (ACT) **GNU**
 DL/I **DBA**
 generation **GNU**
 assembly listing **GNU**
 control statement listing **GNU**
 diagnostics **GNU**
 output **GNU**
 requirements **DBA**
 job control for creating **GNU**
 load module **GNU**
 online operation **DBA**
 CICS/VS partition **DBA**
 generation requirements **DBA**
 storage layout control option **DBA**
 with CICS/VS **DBA**
 interaction with DL/I **DBA**
 online operation programs
 online initialization **DIAG**
 online interface **DIAG**
 online program request handler **DIAG**
 online program examples **CALL**
 Assembler command language environment **CALL**
 Assembler macro language environment **CALL**
 COBOL command language environment **CALL**
 COBOL macro language environment **CALL**
 PL/I command language environment **CALL**
 PL/I macro language environment **CALL**
 RPG II **CALL**
 online program request handler **DIAG**
 online program structure **GNU**
 COBOL **GNU**
 PL/I **GNU**
 online programming considerations **CALL**
 RPG II **GNU**
 online remote data base aids **DIAG**
 online requirements **RDU**
 UPSI byte settings **RDU**
 online RPG II application example **GNU**
 online sample application job stream **GNU**
 online sample application program, DLZSAM60 **GNU**
 DFHFCT - CICS/VS file control table **GNU**
 DFHPCT - CICS/VS program control table **GNU**
 DFHPPT - CICS/VS processing program table **GNU**
 job control **GNU**
 screen formats **GNU**
 online sample application program, DLZSAM80 **GNU**
 DL/I online nucleus generation **GNU**
 online sample application, DL/I **GNU**
 online scheduling trace **DIAG**
 online services of DL/I, requesting **CALL**
 online system **GNU**
 environment **DBA**
 controlling **DBA**
 deferred-open option **DBA**
 special DL/I calls **DBA**
 initialization of **GNU**
 storage, DL/I **DBA**
 online test program (DLZMDL10) **DIAG**
 CICS/VS PCT entry **DIAG**
 CICS/VS PPT entries **DIAG**
 TCA status codes **DIAG**
 online trace
 D(X'C4') type request **DIAG**
 S(X'E2') type request **DIAG**
 T(X'E3') type request **DIAG**
 online trace facilities **DIAG**
 entry ID field **DIAG**
 online scheduling **DIAG**
 register 14 contents **DIAG**
 request codes **DIAG**
 request type **DIAG**
 resource name **DIAG**
 TCA ID number **DIAG**
 TCA ID number, online trace **DIAG**
 time of day, online trace **DIAG**
 trace entry **DIAG**
 trace information field **DIAG**
 type of request **DIAG**
 online user error aids **DIAG**
 online wait state debugging aids **DIAG**
 online, MPS, and batch differences **GNU**
 ONLINEBH (X'80') **DIAG**
 op in WHERE option **HLPI**
 open/close functions **DBA**
 open/close information, verifying with LISTCAT **DIAG**
 operand coding requirements **RDU**
 LOGICAL data bases **RDU**
 virtual logical child segments **RDU**
 operating characteristics **IRDU**
 cursor position **IRDU**
 data entry items **IRDU**
 data entry panels **IRDU**
 default values **IRDU**
 help panels **IRDU**
 operating environments **GIM, HLPI**
 DL/I batch processing **GIM**
 DL/I online processing **GIM**
 distributed data **GIM**
 Multiple Partition Support (MPS) **GIM**
 operation errors **DIAG**
 operational capabilities **DBA**
 choosing **DBA**
 MPS operation **DBA**
 online operation **DBA**
 processing contention control **DBA**
 operational considerations, program isolation **GNU**
 operational procedures **DBA, LLC**
 data base maintenance **DBA**
 data base recovery **DBA**
 performance monitoring **DBA**
 security **DBA**
 operations, data base user responsibility **GNU**
 operator
 field comparison **GNU**
 in WHERE option **HLPI**
 OPTION operand of DLZTRACE **DIAG**
 OPTION parameter description **DIAG**
 options
 and arguments **HLPI**
 definition **HLPI**
 example **HLPI**
 HLPI use **GNU**
 in syntax **HLPI**
 OR, logical **CALL**
 expression of **CALL**
 order item segment (STCCITM), description **GNU**
 other available DL/I functions **HLPI**
 other special statements **DIAG**
 OUT operand **RDU**
 of BUILD control statement **RDU**
 OUTPUT operand of DLZTRACE **DIAG**
 overfield **GNU**
 considerations for **GNU**
 definition of **GNU**
 overflow area (ESDS) **GNU**
 overlay program restriction **GNU, HLPI**
 overlay structures restrictions **CALL**

overview **RDU**
data base implementation **GNU**
extract facility **RDU**
OVFLW operand coding requirements **RDU**

P

PAIR operand
coding requirements **RDU**
LCHILD statement **GNU**
panels
data entry **IRDU**
help **IRDU**
menu **IRDU**
sample **IRDU**
parameter count **CALL**
use of **CALL**
parameter information requirements **GNU**
for MPS **GNU**
parameter list
DLZTRACE user exit **DIAG**
DLZTRCAL **DIAG**
field level sensitivity **DIAG**
for PL/I applications **DIAG**
program request handler **DIAG**
trace point for user exit **DIAG**
user exit **DIAG**
DSECT **DIAG**
user, pointer to **DIAG**
parameter requirements, system calls **DBA**
CMXT **DBA**
STOP **DBA**
STRT **DBA**
TSTP **DBA**
TSTR **DBA**
parameter statement **GNU, HLPI, RDU**
ASLOG operand **GNU**
for DL/I **CALL, RDU**
application program **RRG**
execution **CALL**
when using PL/I **CALL**
HDBFR operand **GNU**
HSBFR operand **GNU**
ISQL Extract Defines Utility **RDU**
LOG operand **GNU**
TRACE operand **GNU**
parameters for DL/I **DIAG**
continuation statements **DIAG**
entering **DIAG**
parameters, batch initialization **IRDU**
parent **HLPI**
and child segments, connecting **DBA**
definition **ABDD, HLPI**
GET NEXT IN PARENT command **HLPI**
items **LLC**
parent operand **DBA, GNU**
for logical child **DBA**
for virtual logical child **DBA**
SEGM statement **GNU**
for a logical child **GNU**
for a logical parent **GNU**
in logical DBD **GNU**
SENSEG statement **GNU**
PARENT operand coding requirements **RDU**
in SEGM statement **RDU**
HD data bases **RDU**
HDAM data bases **RDU**
HIDAM data bases **RDU**

HISAM data bases **RDU**
HSAM data bases **RDU**
logical child segments **RDU**
LOGICAL data bases **RDU**
SHISAM data bases **RDU**
SHSAM data bases **RDU**
virtual logical child segments **RDU**
in SENSEG statement **RDU**
PARENT operand for logical child **DBA**
parent segment **DBA, GNU, HLPI**
example **DBA, GNU**
maximum number of **GNU**
parent segment
definition of **CALL**
destination **HLPI**
parent/child relationship **GNU**
example of **DIAG**
defining **DBA**
parent, logical **GNU**
parentage **CALL, GNU, HLPI**
after CHKP call **CALL**
after DLET call **CALL**
after GN (GHN) call **CALL**
after GNP (GHNP) call **CALL**
after GU (GHU) call **CALL**
after ISRT (add) call **CALL**
after ISRT (load) call **CALL**
establishing with Get command **GNU**
part table, IUG created **DIAG**
PART tables **IRDU**
partial data base reorganization **DBA, DIAG, GNU**
description **IRDU**
function **DBA**
partial data base reorganization utility **RDU**
identifying areas to be reorganized **RDU**
part 1 - DLZPRCT1 **RDU**
coding example **RDU**
control statements **IRDU, RDU**
description **IRDU, RDU**
job control statements **RDU**
illustration **IRDU**
parameters **IRDU**
sample output report **RDU**
part 2 - DLZPRCT2 **RDU**
coding example **RDU**
control statements **IRDU, RDU**
description **IRDU, RDU**
job control statements **RDU**
illustration **IRDU**
parameter statement **IRDU, RDU**
sample output report **RDU**
partial reorganization function, IUG **DIAG**
partial volume dump/copy **RRG**
partition specification table, DL/I **DBA**
partition table, MPC **DIAG**
parts data base **LLC**
bidirectional pointers **LLC**
bill of material relationships **LLC**
description **LLC**
HIDAM used for **LLC**
intersection data stored **LLC**
restrictions **LLC**
logical relationships **LLC**
rules for establishing **LLC**
physical vs logical **LLC**
segment names **LLC**
structure not fixed **LLC**
use **LLC**
where-used relationship **LLC**
PASS operand coding requirements **RDU**

PASS parameter, DLZACT macro GNU
 passback (MPS) GNU
 patch area DIAG
 path DBA
 hierarchical HLPI
 path call GNU, HLPI
 and REPL call CALL
 command code D CALL
 definition HLPI
 example of GNU
 in INSERT command HLPI
 segment-not-found HLPI
 using the FROM option GNU
 using the INTO option GNU
 PCB (program communication block) CALL, GNU
 address in argument list CALL
 Assembler mask for CALL
 COBOL mask for CALL
 coding the option GNU
 composition (RPG II) GNU
 contents of CALL
 example of GNU
 in PSB CALL
 mask CALL
 option GNU
 PL/I mask for CALL
 schedule a PSB GNU
 specifying a GNU, HLPI
 use of CALL
 multiple CALL
 PCB addressability, online environment CALL
 PCB call, online CALL
 PCB compare DIAG
 PCB entries, access to CALL
 PCB mask CALL
 DBD name CALL
 key feedback area CALL
 length CALL
 number of sensitive segments CALL
 processing options CALL
 RPG II GNU
 segment level CALL
 segment name feedback CALL
 status code CALL
 PCB names, sequence specification CALL
 PCB option HLPI
 PCB statement GNU
 DBDNAME operand GNU
 format RDU
 KEYLEN operand GNU
 POS operand GNU
 PROCOPT operand GNU
 PROCSEQ operand GNU
 PCB structure, defining RPG II GNU
 PCBs in PSB GNU
 PCT (program control table), CICS/VS DBA, GNU, RDU, RRG
 and PPT entries DBA
 PDIR address DIAG
 PDUMP DIAG
 PDZAP DIAG
 performance GNU
 and logging DBA, GNU
 aspects, physical data structure GNU
 CICS/VS GNU
 considerations DBA
 file control table (FCT) GNU
 randomized access (HD) GNU
 performance considerations CALL, LLC
 GN call usage CALL
 in system view ADBD
 procedure ADBD
 multiple positioning CALL
 number of buffer subpools CALL
 online environment CALL
 qualified SSAs CALL
 when adding segments CALL
 periodic backups RRG
 personnel DBA
 DL/I data base utilities DBA
 requirements DBA
 real DBA
 virtual DBA
 roles DBA
 application design DBA
 application programming DBA
 data base administration DBA
 data base planning and design DBA
 system operation DBA
 system programming DBA
 training DBA
 personnel requirements ADBD
 PERT chart GNU
 example DBA
 PF keys IRDU
 PGMNAME
 operand coding requirements RDU
 parameter, DLZACT macro GNU
 phase 1 GNU
 sample application, description GNU
 phase 2 GNU
 logical data bases GNU
 logical DBD
 for the customer data base GNU
 for the inventory data base GNU
 physical data bases GNU
 physical DBDs GNU
 phase 3 GNU
 physical data bases GNU
 physical DBDs GNU
 physical access path GNU
 physical child
 backward pointer DBA
 first pointer DBA
 last pointer DBA
 pointer GNU
 physical data bases ADBD, GNU, HLPI
 structure GNU
 and access methods GNU
 physical data structure, designing GNU
 physical DBD GNU
 accessing a logical child in GNU
 coding a logical relationship in GNU
 examples GNU
 with logical relationships, examples GNU
 physical delete rule violation CALL
 physical insert rule violation CALL
 physical parent
 (bidirectional only), defining the sequence of DBA
 how to define in a logical relationship RDU
 insert rules (bidirectional relationships only) DBA
 pointer (PP) GNU
 replace rules (bidirectional relationships only) DBA
 segment GNU
 type GNU
 physical path DBA
 physical pointer summary for HD data bases DIAG
 physical record, segment relationship CALL
 physical replace rule violation CALL
 physical storage device type DBA
 physical structures, defining DBA

physical twin backward pointer **DBA**
 physical twin forward **DBA**
 physical twin pointer **GNU**
 physical twin pointer options **DBA**
 physical twin pointers, suppression of **DBA**
 physical versus logical relationships **DBA**
 PI (see program isolation) **DIAG, GNU**
 PI operand coding requirements **RDU**
 PI support testing **DIAG**
 PITRACE (X'90') **DIAG**
 PL/I **GNU**
 and SEGLENGTH option **GNU**
 batch program structure **GNU**
 entry to an application program **GNU**
 online job control **GNU**
 online program structure **GNU**
 sample program names **GNU**
 PL/I (see application programs, high-level languages) **LLC**
 PL/I application program **CALL, GNU**
 batch program example **CALL**
 combined checkpoint example **CALL, GNU**
 DL/I call format **CALL**
 entry points **CALL**
 error processing example **CALL**
 examples of online calls **CALL**
 job control for compile and link **CALL**
 link editing of **CALL**
 online call **CALL**
 scheduling call **CALL**
 termination call **CALL**
 PCB mask **CALL**
 termination of **CALL**
 PL/I batch control flow **DIAG**
 PL/I batch program **CALL**
 restrictions **CALL**
 structure **CALL**
 PL/I call **DIAG**
 PL/I diagnostic information **HLPI**
 PL/I language **CALL**
 restrictions **CALL**
 PL/I MPS batch control flow **DIAG**
 PL/I online control flow (CICS/VS) **DIAG**
 PL/I parameter list **DIAG**
 locating the **DIAG**
 planning considerations **RRG**
 considerations **RRG**
 for recovery **RRG**
 for recovery/restart **RRG**
 backup/restore **RRG**
 project approach to **DBA**
 your program **HLPI**
 PLICALLB entry point **DIAG**
 PLITDLI **DIAG**
 PLT (program list table) **DBA, GNU, RDU**
 CICS/VS **GNU**
 contents **GNU**
 PLU parameter information **DIAG**
 pointer
 affect on performance **ADBD**
 dependent segments chained by **ADBD**
 fast access to path of segments **ADBD**
 in bidirectional logical relationships **ADBD**
 in direct access **ADBD**
 in indexed access **ADBD**
 in indexing **ADBD**
 in logical relationships **ADBD**
 maintained internally **ADBD**
 not used in sequential organization **ADBD**
 overhead **ADBD**
 placement for logical relationships **ADBD**
 segment **ADBD**
 special in indexed access **ADBD**
 to free space **ADBD**
 POINTER operand **GNU**
 SEGM statement **GNU**
 SEGM statement for a logical child **GNU**
 SEGM statement for a logical parent **GNU**
 summary of keyword options **RDU**
 POINTER operand coding requirements **RDU**
 in LCHILD statement **RDU**
 index target segments in HDAM and HIDAM data bases **RDU**
 logical child segments **RDU**
 primary INDEX data bases **RDU**
 primary index relationships in HIDAM data bases **RDU**
 secondary INDEX data bases **RDU**
 in SEGM statement **RDU**
 HD data bases **RDU**
 HIDAM data bases **RDU**
 HIDAM data bases **RDU**
 logical child segments **RDU**
 virtual logical child segments **RDU**
 pointer options, defining additional **DBA**
 pointer segment **ADBD**
 pointer, position **HLPI**
 in GET commands **HLPI**
 pointers **GNU**
 (see POINTER and PARENT operands) **RDU**
 direct address **DIAG**
 for HIDAM data base **DIAG**
 generated by DBDGEN **DIAG**
 in an HD data base **GNU**
 in segment flag **DIAG**
 options generated **DIAG**
 physical child **GNU**
 physical twin **GNU**
 specifying **DIAG**
 twin backward **DIAG**
 used for logical relationships **GNU**
 value formula **DIAG**
 pointers for HD data bases **DIAG**
 pointers, optional for logical relationships **DBA**
 pool control, HDBFR **DBA**
 POS operand
 coding requirements **RDU**
 PCB statement **GNU**
 position in a data base, establishing **GNU**
 position pointer **CALL, HLPI**
 after CHKP call **CALL**
 after DLET call **CALL**
 after GN (GHN) call **CALL**
 after GNP (GHNP) call **CALL**
 after GU (GHU) call **CALL**
 after ISRT (add) call **CALL**
 after ISRT (load) call **CALL**
 in GET commands **HLPI**
 positioning **GNU, HLPI**
 current position **GNU**
 data base **GNU**
 data base after a call **GNU**
 basic rules **GNU**
 multiple **HLPI**
 rules for after an HLPI command **GNU**
 single **HLPI**
 potential users of DL/I **GNU**
 PP (physical parent pointer) **GNU**
 PPST
 chain address **DIAG**
 chaining **DIAG**
 PPT (processing program table) **DBA, GNU, RDU**
 and PCT entries **DBA**

pre-installation planning **DBA**
 application design **DBA**
 data base design **DBA**
 definition **DBA**
 project approach **DBA**
 gross PERT chart **DBA**
 project cycle **DBA**
 sample project plan **DBA**
 user responsibilities **DBA**
 data base design **DBA**
 machine requirements **DBA**
 operational procedures **DBA**
 personnel **DBA**
 programming system requirements **DBA**
 scheduling **DBA**
 system implementation **DBA**
 user-written programming **DBA**
 precautions, split screen **IRDU**
 prefix resolution **DBA**
 data base **DBA**
 prefix resolution function, IUG **DIAG**
 prefix resolution utility (DLZURG10) **DIAG, GNU, RRG**
 prefix update **DBA**
 data base **DBA**
 function, IUG **DIAG**
 utility (DLZURGP0) **DIAG, GNU, RRG**
 preliminary analysis **ABDD**
 adding an application **ABDD**
 how to use for first application **ABDD**
 interest from end users **ABDD**
 setting up a team **ABDD**
 support of management **ABDD**
 initial application **ABDD**
 interest from end users **ABDD**
 setting up a team **ABDD**
 support of management **ABDD**
 preparing for program execution **DBA**
 batch partition controller **DBA**
 batch reinitialization **DBA**
 batch system environment **DBA**
 illustrated **DBA**
 multiple partition support operation **DBA**
 illustrated **DBA**
 operating under CICS/VS **DBA**
 requirements for execution **DBA**
 restart facility **DBA**
 system requirements **DBA**
 preparing to implement data base **DBA**
 choosing DL/I definitional capabilities **DBA**
 access techniques **DBA**
 data base structures **DBA**
 data formats **DBA**
 processing controls **DBA**
 choosing DL/I operational capabilities **DBA**
 buffer pool assignment **DBA**
 distributed data **DBA**
 limited data sharing **DBA**
 logging **DBA**
 MPS operation **DBA**
 online operation **DBA**
 processing contention control **DBA**
 real storage assignment **DBA**
 choosing the access method **DBA**
 access method characteristics **DBA**
 defining physical characteristics **DBA**
 CI and block sizes **DBA**
 distributed free space **DBA**
 logical record length **DBA**
 physical storage device type **DBA**
 scanning for storage space **DBA**
 VSAM options **DBA**
 VSAM space requirements **DBA**
 input **DBA**
 preparing to use DL/I **CALL, HLPI**
 prereorganization **DBA**
 function, IUG **DIAG**
 utility (DLZURPR0) **DIAG, GNU, RRG**
 primary access, types of **GNU**
 primary area (ESDS) **GNU**
 primary index **CALL**
 primary INDEX data bases **RDU**
 coding example **RDU**
 control statements, format **RDU**
 DATASET **RDU**
 DBD **RDU**
 DBDGEN **RDU**
 END **RDU**
 FIELD **RDU**
 FINISH **RDU**
 LCHILD **RDU**
 SEGM **RDU**
 control statements, summary **RDU**
 how to define **RDU**
 (see also HIDAM data bases) **RDU**
 input structure and rules **RDU**
 primary index reorganization **DBA**
 primary index specification in the HIDAM data base **DBA**
 primary index, HD **DBA**
 primary key
 candidate **ABDD**
 definition **ABDD**
 primary menu
 menu, primary **IRDU**
 primary processing sequence **CALL**
 PRIMCI operand coding requirements **RDU**
 printed report **RDU**
 from ISQL query routine **RDU**
 priority assignment **DBA**
 problem determination **CALL, GNU, HLPI**
 description **IRDU**
 execution errors **CALL**
 high level language debugging **CALL**
 initialization errors **CALL**
 IUG **GNU**
 status codes **CALL**
 detail **CALL**
 summary **CALL**
 problem determination dumps **DIAG**
 log print **DIAG**
 trace print **DIAG**
 job control statements **DIAG**
 problem determination utilities
 trace print **IRDU**
 problem reporting **DIAG**
 APAR requirements **DIAG**
 documentation requirements **DIAG**
 PROC names, entering **IRDU**
 procedures **LLC**
 audit **LLC**
 control **LLC**
 operational **LLC**
 reconstruction **LLC**
 process controls
 for fields, defining **DBA**
 for segments, defining **DBA**
 PROCESS statement **HLPI**
 processes
 IMF **IRDU**
 IUG **IRDU**
 process industry **GIM**

- online order entry and production control system **GIM**
- processing **GNU**
 - contention control **DBA**
 - controls **DBA**
 - data bases with RPG II **GNU**
 - logically related segments **GNU**
 - options **DBA**
 - reorganizing a HISAM data base **DBA**
 - reorganizing a primary or secondary index **DBA**
 - reorganizing a SHISAM data base **DBA**
 - reorganizing an HD data base **DBA**
 - reorganizing an HD, HDAM or HIDAM data base **DBA**
 - sequence, secondary **GNU**
- processing authority **DBA**
 - controlling **ADBD, DBA**
- processing considerations, data base **HLPI**
- processing contention control **DBA**
- processing controls **DBA**
- processing description **LLC**
 - control data base **LLC**
 - segment, LLCTL **LLC**
 - segment, PARTEXP **LLC**
 - segment, UPDMASTER **LLC**
- exploding technique **LLC**
 - example **LLC**
- hierarchical path **LLC**
 - defined **LLC**
 - following discontinued **LLC**
 - processing terminated **LLC**
- initial generation mode **LLC**
 - abnormal conditions **LLC**
 - control data base **LLC**
 - no processing required, reason **LLC**
- updating mode **LLC**
 - component key **LLC**
 - parent key **LLC**

- processing methods, data base **HLPI**
- processing option **CALL**
- data base load **CALL**
- definition of **CALL**
- in PCB mask **CALL**
- processing option (PROCOPT) **DBA, GNU**
- delete processing **DBA**
- get processing **DBA**
- GO, programming considerations **GNU**
- insert processing **DBA**
- load **DBA**
 - sequentially **DBA**
- P **GNU**
- PROCOPT=GO(P) **DBA**
- PSB PROCOPT selection **DBA**
- read-only **DBA**
- replace processing **DBA**
- types **DBA**
 - all (A) **DBA**
 - delete (D) **DBA**
 - exclusive use (E) **DBA**
 - get (G) **DBA**
 - insert (I) **DBA**
 - path (P) **DBA**
 - replace (R) **DBA**
- processing option, PROCOPT=GO(P) **DIAG, HLPI**
- processing program table (see PPT) **DBA, RDU**
- processor capability requirements **ADBD**
- PROCOPT **DIAG, GNU**
- guidelines **GNU**
- operand
 - PCB statement **GNU**
 - SENSE statement (PSBGEN) **GNU**
- parameter **GNU**
 - selection, PSB **GNU**
 - use in loading **GNU**
- PROCOPT (see processing option) **CALL**
- operand coding requirements **RDU**
 - PCB statement **RDU**
 - SENSE statement **RDU**
- processing option **DIAG**
- PROCSEQ operand **GNU**
- in PSBGEN **CALL**
- PCB statement **GNU**
- secondary index, PCB statement (PSBGEN) **GNU**
- product structure tree **LLC**
- productivity **GIM**
- application programmer productivity **GIM**
- application programming languages **GIM**
- data and device independence **GIM**
- user aids **GIM**
- DB design aids **GIM**
- Data dictionary **GIM**
- progname parameter **RDU**
- program **GNU**
- communication block (PCB) **GNU**
- control table (PCT), CICS/VS **GNU**
- execution, DL/I **GNU**
- list table (PLT), CICS/VS **GNU**
- processing table (PPT), CICS/VS **GNU**
- specification block (PSB) **GNU**
 - remote **GNU**
 - generation (PSBGEN) **GNU**
- structure, DL/I **GNU**
- program check **DIAG**
- program check handler, CICS/VS **DIAG**
- program communication block (PCB) **CALL, DBA, CALL, HLPI**
- program control table (PCT), CICS/VS **RRG, DBA, RDU**
- program debugging **HLPI**
- program examples **CALL**
- Assembler batch program structure **CALL**
- COBOL batch program structure **CALL**
- PL/I batch program structure **CALL**
- RPG II batch program structure **CALL**
- program execution **DBA**
- batch **DBA**
 - data base system flow **DBA**
 - system control flow **DBA**
 - system execution **DBA**
- MPS batch **DBA**
- online **DBA**
 - functions **DBA**
 - initialization of CICS/VS **DBA**
 - system control flow illustrated **DBA**
 - system execution **DBA**
- program execution, batch and MPS batch application **CALL**
- program function (PF) keys **IRDU**
- program isolation (PI) **DBA, DIAG, GNU**
- and intent scheduling **DBA**
- CICS/VS abnormal termination **GNU**
- comparison with intent scheduling **GNU**
- concurrent updates **GNU**
- contention management **GNU**
- deadlock avoidance **GNU**
- DLZACT macro specification **GNU**
- exclusive intent **GNU**
- integrity **GNU**
- operational considerations **GNU**
- parameter, DLZACT macro **GNU**
- programming considerations **GNU**
- read-only intent **GNU**
- update intent **GNU**
- program list table (see PLT), CICS/VS **DBA, RDU**
- program packages, use of **DBA**

program processing table (PPT), CICS/VS **DBA**
 program request handler **DIAG**
 MPS batch **DIAG**
 online **DIAG**
 parameter list **DIAG**
 program requirements, application **DBA**
 program specification block (PSB) **CALL, DBA, GNU, HLPI**
 definition **HLPI**
 generation (see PSBGEN) **GNU, IRDU**
 remote **GNU**
 specifying **HLPI**
 program storage allocation **DBA**
 program structure **GNU**
 and interface to DL/I **GNU**
 RPG II and interface to DL/I **GNU**
 program termination **GNU**
 batch **GNU**
 MPS batch **GNU**
 program test requirements **DBA**
 application program requirements **DBA**
 data base requirements **DBA**
 DL/I test program **DBA**
 sample data requirements **DBA**
 test requirements **DBA**
 programming considerations **DBA, GNU**
 CICS/VS transactions **GNU**
 MPS **GNU**
 processing option GO **GNU**
 program isolation **GNU**
 programming examples **HLPI**
 programming hints, application **DBA**
 programming languages, DL/I supported **GNU**
 programming suggestions, general **CALL**
 programming systems **LLC**
 programming techniques **CALL**
 programming techniques and suggestions **HLPI**
 programming, optional DL/I **CALL**
 command codes **CALL**
 DL/I system calls **CALL**
 field level sensitivity **CALL**
 GSCD calls **CALL**
 multiple positioning **CALL**
 secondary indexing **CALL**
 variable length segments **CALL**
 programming, user-written **DBA**
 project approach to data base implementation **GNU**
 project cycle **DBA, GNU**
 data base implementation **GNU**
 manpower requirements **DBA**
 phases **DBA**
 sample project plan **DBA**
 Gantt chart **DBA**
 PERT chart **DBA**
 project plan, sample **DBA**
 PSB **ADBD, HLPI**
 data gathering **DIAG**
 default table **DIAG**
 definition table **DIAG**
 generation **DIAG**
 intent scheduling **DIAG**
 LIFO list **DIAG**
 module flow **DIAG**
 naming convention **ADBD**
 obtaining a **HLPI**
 releasing a **HLPI**
 scheduling intent **DIAG**
 tables **DIAG**
 PSB (program specification block) **CALL, DBA, CALL, GNU, RDU**
 basic PSBs **RDU**
 coding example **RDU**
 control statements, format **RDU**
 control statements, summary **RDU**
 input structure and rules **RDU**
 building **CALL**
 creating interactively **GNU**
 for the online application, update **GNU**
 generation **DBA**
 how to define **RDU**
 how to generate **RDU**
 illustrated **DBA**
 load customer and inventory data bases phase 3 **GNU**
 logical customer data base **GNU**
 phase 2 **GNU**
 logical inventory data base **GNU**
 phase 2 **GNU**
 naming convention **DBA**
 operand, BUILD statement (PSBGEN) **GNU**
 PROCOPT selection **GNU**
 PSBs for loading data bases **RDU**
 for loading data bases **RDU**
 coding example **RDU**
 for logical data bases **RDU**
 coding example **RDU**
 for secondary indexes **RDU**
 coding example **RDU**
 with field level sensitivity **RDU**
 control statements, format **RDU**
 control statements, summary **RDU**
 remote **GNU**
 scheduling **GNU**
 storage **DBA**
 storing information into SQL/DS tables **RDU**
 to process customer data base **GNU**
 use to determine data base organization **GNU**
 PSB default table **IRDU**
 PSB definition **IRDU**
 table **IRDU**
 PSB generation **DIAG, IRDU**
 CMS macro file **IRDU**
 definition **IRDU**
 diagnostic messages **RDU**
 generation **IRDU**
 job control statements **RDU**
 output listing **RDU**
 procedure **IRDU, RDU**
 VSE job stream **IRDU**
 PSB macros relating to IMF panels **IRDU**
 PSB operand **RDU**
 coding requirements **RDU**
 of BUILD control statement **RDU**
 PSBs **GNU**
 option **HLPI**
 tables **DIAG**
 used for the phase 3 sample application **GNU**
 worksheets **IRDU**
 PSBBASICDATA **RDU**
 contents **RDU**
 description **DBA, RDU**
 PSBFIELD **RDU**
 PSBFIELDDATA **RDU**
 contents **RDU**
 description **DBA, RDU**
 PSBGEN (program specification block generation) **GNU, RDU**
 definition **IRDU, RDU**
 description **IRDU**
 execution, job control **GNU**
 generation **IRDU, RDU**
 IMF **GNU**
 input deck structure **GNU**
 macros **DBA**

output, description **GNU**
 secondary index PCB statement **GNU**
 SENFLD statement **GNU**
 SENSEG statement **GNU**
 statement
 format **RDU**
 LANG operand **GNU**
 PSBNAME operand **GNU**
 VIRFLD statement **GNU**
PSBNAME DIAG
 operand coding requirements **RDU**
 DLZACT TYPE=PROGRAM statement **RDU**
 PSBGEN statement **RDU**
PSBNAME parameter GNU, RDU
 DLZACT macro **GNU**
 PSBGEN statement **GNU**
PSBNAMES RDU
PSBPCB RDU
PSBPCBDATA RDU
 contents **RDU**
 description **DBA, RDU**
PSBSEG RDU
PSBSEGMENTDATA RDU
 contents **RDU**
 description **DBA, RDU**
PSBSEGV RDU
PSTBYTNM DIAG
PSTSV1, register save area DIAG
PTF number, identifying in dumps DIAG
PTR (see POINTER operand) RDU
 punch statement format **DIAG**

Q

Q command code GNU
QSSA (qualified SSA) GNU
 format of **GNU**
 qualification on non-key fields **GNU**
 qualification statement (SSA) **CALL, GNU**
 comparative value **CALL**
 new set of **CALL**
 relational operator **CALL**
 segment field name **CALL**
 termination of **CALL**
 use of **CALL**
 qualified segment selection **GNU, HLPI**
 boolean operators **HLPI**
 in WHERE option **HLPI**
 example of **GNU**
 root non-key field warning **HLPI**
 WHERE option **GNU, HLPI**
 query routines, ISQL **RDU**
 for Documentation Aid facility **RDU**
 sample **RDU**

R

random retrieval **HLPI**
 randomized access **DBA**
 characteristics **ABDD, DBA**
 how it works **ABDD**
 chain of root segments **ABDD**
 free space **ABDD**
 HD options **ABDD**
 physical block number **ABDD**
 randomizing routine **ABDD**

 root anchor point **ABDD**
 synonyms **ABDD**
HD GNU
 data bases **GNU**
 overview of **GNU**
 performance **GNU**
 method of data storage **ABDD**
 performance **ABDD**
 uses **ABDD**
 randomized module
 use of for HD data bases **GNU**
 errors **DIAG**
 randomizing module examples **DBA**
 randomizing module interfaces **DBA**
 randomizing module, where defined **RDU**
 randomizing modules **DBA**
 anchor point number **DBA**
 binary halving method **DBA**
 illustrated **DBA**
 compiling and testing **DBA**
 control blocks **DBA**
 conversion **DBA**
 examples **DBA**
 binary halving method **DBA**
 division **DBA**
 hashing method **DBA**
 modulo **DBA**
 field value parameter **DBA**
 generalized HDAM example **DBA**
 register contents, entry **DBA**
 hashing method **DBA**
 illustrated **DBA**
 HD randomized data bases **DBA**
 interfaces **DBA**
 invoking **DBA**
 loading **DBA**
 modulo or division method **DBA**
 illustrated **DBA**
 naming **DBA**
 primary input parameters **DBA**
 randomizing module - generalized HDAM **DBA**
 register contents, entry **DBA**
 relative block number **DBA**
 segment selection argument **DBA**
 randomizing routine **ABDD**
 for sequential access **ABDD**
 direct access by **DBA**
RAP (root anchor point) GNU
RBA (relative byte address) DBA
 read-only intent **DBA**
 real storage **GIM**
 real storage estimates, DL/I **DBA**
 real storage requirements **ABDD**
 example, DL/I **DBA**
Reassembling DL/I for new/updated CICS/VS DBA
 buffer pool assignment **DBA**
 HD access methods **DBA**
 HS access methods **DBA**
 distributed data **DBA**
 extended remote PSB **DBA**
 remote PSB **DBA**
 limited data sharing **DBA**
 logging **DBA**
 asynchronous logging **DBA**
 CICS/VS system journal **DBA**
 DL/I system log **DBA**
 MPS operation **DBA**
 CICS/VS file control table **DBA**
 CICS/VS journal control table **DBA**

- CICS/VS program control table **DBA**
- CICS/VS program list table **DBA**
- CICS/VS program processing table **DBA**
- CICS/VS system initialization table **DBA**
- DL/I application control table **DBA**
- processing contention control **DBA**
- recommended naming convention **ADBD**
- additional for DL/I **ADBD**
- DBD **ADBD**
- field **ADBD**
- file-id **ADBD**
- filename **ADBD**
- job name **ADBD**
- phase name **ADBD**
- program name **ADBD**
- PSB **ADBD**
- segment **ADBD**
- reconstruction procedures **LLC**
- record **ADBD**
- length, logical **DBA**
- level enqueueing **DBA**
- RECORD operand
- coding requirements **RDU**
- HISAM data bases **RDU**
- HSAM data bases **RDU**
- primary INDEX data bases **RDU**
- secondary INDEX data bases **RDU**
- SHISAM data bases **RDU**
- SHSAM data bases **RDU**
- DATASET statement **GNU**
- records
- controlling processing of **DBA**
- data base **GNU**
- defining sequencing for **DBA**
- direct processing **DBA**
- sequencing **DBA**
- recovery **GNU**
- concepts **RRG**
- considerations, VSAM **GNU**
- data base **DBA, GNU**
- decision chart
- for an online system **RRG**
- for batch **RRG**
- for MPS batch **RRG**
- description **IRDU**
- file **GNU**
- for online users **RRG**
- from backed up data **RRG**
- off-line **DBA**
- online **GNU**
- procedure **GNU**
- documentation **GNU**
- flowchart **GNU**
- utilities **GNU**
- DL/I **GNU**
- data set **GNU**
- VSAM considerations **GNU**
- recovery failures **RRG**
- during backout **RRG**
- during data base restore **RRG**
- recovery function, IUG recovery **DIAG**
- recovery module flow, IUG **DIAG**
- recovery of DL/I-IMS compatible data bases **DBA**
- recovery planning **DBA, RRG**
- basic choices **RRG**
- considerations for **RRG**
- reasons for **RRG**
- strategy **RRG**
- trade-offs **RRG**
- recovery procedures, normal **RRG**

- recovery requirements **ADBD**
- recovery strategy **RRG**
- considerations for **RRG**
- DL/I offerings **RRG**
- mixing use of **RRG**
- selecting one **RRG**
- recovery tools
- logging facility **DBA**
- asynchronous logging option **DBA**
- description **DBA**
- force writes **DBA**
- logging and performance **DBA**
- minimizing writing to log **DBA**
- write ahead logging **DBA**
- recovery utilities **RDU, RRG**
- backout **DIAG**
- change accumulation **DIAG**
- data base backout **IRDU**
- data base change accumulation **IRDU**
- data base data set image copy **IRDU**
- data set recovery **DIAG, IRDU**
- description **IRDU**
- image copy **DIAG**
- list **IRDU**
- log print **DIAG, IRDU**
- recovery/restart, reasons for **RRG**
- recovery/restart, VSAM considerations **DBA**
- recreating a data base **RRG**
- redundancies
- eliminating **ADBD**
- REF operand
- coding requirements **RDU**
- HD primary indexed access **RDU**
- HD secondary indexed access **RDU**
- primary indexed access **GNU**
- secondary index, ACCESS statement **GNU**
- reference, syntax **HLPI**
- register contents on entry **DBA**
- register save areas **CALL, DIAG**
- locating **DIAG**
- regression testing **DIAG**
- reinitialization of batch environment **CALL, GNU, HLPI**
- during MPS restart **CALL, GNU**
- relational operator **GNU**
- in qualified SSA **CALL**
- operator, SSA **GNU**
- relationships **ADBD**
- many-to-many **ADBD**
- many-to-one **ADBD**
- one-to-many **ADBD**
- one-to-one **ADBD**
- relationships
- bidirectional **DBA**
- logical (see logical relationships) **GNU**
- relative byte address (RBA) **DBA**
- releasing a PSB (TERMINATE) **GNU**
- releasing a PSB in a CICS/VS application **GNU**
- releasing the PSB **HLPI**
- reload program, using your own **DBA**
- remote data bases **DIAG**
- aids **DIAG**
- interfaces **DIAG**
- remote interface block (RIB) **DIAG**
- REMOTE operand coding requirements **RDU**
- REMOTE parameter, DLZACT macro **GNU**
- remote PSB **DBA, GNU**
- extended **DBA**
- remote PSB extended **DIAG**
- removing data (see DELETE command) **HLPI**
- reorganization **GNU**

choosing the right utilities **DBA**
 data base **GNU**
 prefix resolution **DBA**
 prefix update **DBA**
 preorganization **DBA**
 scan **DBA**
 description **IRDU**
 frequency **GNU**
 HD data bases **GNU**
 HD reorganization reload **DBA**
 HD reorganization unload **DBA**
 HISAM reorganization reload **DBA**
 HISAM reorganization unload **DBA**
 logical **GNU**
 overview **GNU**
 partial data base reorganization **DBA**
 physical **GNU**
 procedures **DBA**
 programs **GNU**
 protecting your data base **DBA**
 reasons for reorganization **DBA**
 reload
 HD DBA
 HISAM DBA
 reorganization sequence **DBA**
 sequence **DBA**
 steps in reorganizing **DBA**
 unload and reload utilities **DBA**
 unload, HISAM **DBA**
 updating segment prefixes **DBA**
 utilities **GNU**
 using the **DBA**
 what is it **GNU**
 when to reorganize **DBA, GNU**
 reorganization procedures **DBA**
 reorganization reload (HISAM) **RRG**
 Reorganization Reload utility (HD) **RRG**
 reorganization requirements **ADBD**
 reorganization unload (HISAM) **RRG**
 Reorganization Unload utility (HD) **RRG**
 Reorganization Unload utility (HISAM) **RRG**
 reorganization utilities **DBA, DIAG**
 by menu option and function **IRDU**
 choosing the right utilities **DBA**
 data base prefix
 resolution **IRDU**
 update **IRDU**
 data base preorganization **IRDU**
 data base scan **IRDU**
 description **IRDU**
 generating job streams **IRDU**
 HD reorganization reload **DIAG, IRDU**
 HD reorganization unload **DIAG, IRDU**
 HISAM reorganization reload **DIAG, IRDU**
 HISAM reorganization unload **DIAG, IRDU**
 list **IRDU**
 partial data base reorganization **DIAG, IRDU**
 sequence **IRDU**
 unload and reload utilities **DBA**
 reorganization/load **GNU**
 flowchart **GNU**
 processing **GNU**
 reorganization/load module flow (IUG) **DIAG**
 reorganization
 insert function **CALL**
 partial **GNU**
 reorganize, when you should **DBA**
 reorganizing data bases **DBA**
 description **IRDU**
 HD DBA
 HDAM DBA
 HIDAM DBA
 HISAM DBA
 primary index **DBA**
 secondary index **DBA**
 to add a logical relationship **DBA**
 reorganizing a HISAM or SHISAM data base **DBA**
 reorganizing, steps in **DBA**
 repeating work **RRG**
 REPL (replace) **DIAG, GNU**
 call **CALL**
 detailed description **CALL**
 request **DIAG**
 with path call **CALL**
 REPLACE command **GNU, HLPI**
 format of **GNU**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 rule for logical child segment **GNU**
 rule for logical parent segment **GNU**
 rule for physical parent segment **GNU**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 use with Get command **GNU**
 replace - REPL **DIAG**
 replace call path **DIAG**
 delete/replace **DIAG**
 replace **DIAG**
 REPLACE operand coding requirements **RDU**
 replace rules **DBA**
 logical parent **DBA**
 physical parent **DBA**
 replacement rule (see RULES operand) **RDU**
 logical relationships **GNU**
 replacing **HLPI**
 a segment, example **CALL**
 segments **GNU**
 in a data base **GNU**
 with a secondary index **GNU**
 reporting problems **DIAG**
 requirements **GNU, LLC**
 for installation of **LLC**
 DL/I bridge **LLC**
 initial generation **LLC**
 updating **LLC**
 initialization **GNU**
 job control language **GNU**
 of application programs, Assembler language **LLC**
 of control data base initialization **LLC**
 of modules for link-editing initialization program for
 control data base **LLC**
 of user-written application program **LLC**
 parameter information **GNU**
 requirements for execution **DBA**
 application control blocks **DBA**
 creation and maintenance **DBA**
 documentation aid illustrated **DBA**
 illustrated **DBA**
 application programs **DBA**
 data base space allocation **DBA**
 DBD generation **DBA**
 illustrated **DBA**
 requirements listing
 definition **ADBD**
 for sample application **ADBD**
 knowledge about data elements **ADBD**
 source of input requirements **ADBD**

source of output requirements **ABDD**
 use in implementation plan **ABDD**
 using to create data dictionary **ABDD**
 using to define tasks **ABDD**
 reserved
 field in PCB **CALL**
 flag bit indicating wait for resource **CALL**
 keywords **HLPI**
 labels **GNU, HLPI**
 resolution utilities, overview **GNU**
 resolving problems, description **IRDU**
 resource name, online trace **DIAG**
 response to commands **HLPI**
 to commands **HLPI**
 to a call (RPG II) **GNU**
 restart **DBA, GNU, RRG**
 batch jobs **RRG**
 concepts of **RRG**
 considerations **DBA, GNU**
 data base **GNU**
 facility
 for MPS programs **CALL, GNU, HLPI**
 MPS batch **RRG**
 online jobs **RRG**
 procedure **GNU**
 programs, user-written **DBA**
 VSAM considerations **GNU**
 restarting a program from a checkpoint **RRG**
 batch processing **RRG**
 MPS batch processing **RRG**
 restarting an MPS batch program **CALL**
 considerations **CALL**
 job control example **CALL**
 restarting an MPS restart program **GNU**
 JCL example **GNU**
 restarting in MPS batch programs **DBA, HLPI**
 restoring a data base **RRG**
 from a backup copy **RRG**
 Data Set Image Copy Backup **RRG**
 Fast Copy Disk/Copy Disk Volume Backup **RRG**
 HD reorganization unload backup **RRG**
 HISAM reorganization unload backup **RRG**
 restoring backed up data **RRG**
 restoring data base (see CHECKPOINT command) **HLPI**
 retrieving data **RDU**
 from DL/I data base **RDU**
 using Extract Defines **RDU**
 restrictions **GNU, HLPI, RDU**
 CICS/VS **GNU**
 COMREG **GNU**
 labels **GNU**
 MPS batch **HLPI**
 on host language use **GNU**
 on host language use and features **HLPI**
 on mixed use of interfaces **GNU, HLPI**
 on use of
 COMREG **HLPI**
 overlay programming **HLPI**
 reserved keywords and labels **HLPI**
 set exit abnormal (STXIT AB) linkage **HLPI**
 on using logical relationship resolution utilities **RDU**
 online **HLPI**
 overlay programs **GNU**
 reserved keywords **GNU**
 set exit abnormal linkage **GNU**
 SSA length **DIAG**
 STXIT **GNU**
 restrictions on
 change of segment layout or names in control data base **LLC**
 storing of intersection data in parts data base **LLC**

resume interrupted job **GNU**
 description **IRDU**
 IUG **GNU**
 resume processing **IRDU**
 retrieval **HLPI**
 random **HLPI**
 sequential **HLPI**
 retrieval of root segment **CALL**
 retrieval of segments **CALL**
 with and without SSAs **CALL**
 with path call **CALL**
 with qualified SSAs **CALL**
RETRIEVE (X'20') **DIAG**
 retrieve call path **DIAG**
 retrieve calls, inbound **DIAG**
 retrieve subroutine trace **DIAG**
 retrieving **HLPI**
 DL/I documentation aid job streams **RDU**
 ISQL Extract Defines job stream **RDU**
 key feedback **HLPI**
 segments **GNU**
 with a secondary index **GNU**
RETURN (PL/I) **CALL, GNU**
 statement **DIAG**
 return code checking
 DLZALTS **DIAG**
 DLZBH **DIAG**
 return codes **LLC**
 application program test for **LLC**
 use of **LLC**
 returning control to DL/I **GNU**
 returns **DBA**
 abnormal **DBA**
 CMXT **DBA**
 STOP **DBA**
 STRT **DBA**
 TSTP **DBA**
 TSTR **DBA**
 normal **DBA**
 CMXT **DBA**
 STOP **DBA**
 STRT **DBA**
 TSTP **DBA**
 TSTR **DBA**
RIB address **DIAG**
RILIM operand coding requirements **RDU**
RMNAME operand coding requirements **RDU**
RMRTN operand
 coding requirements **RDU**
 primary randomized access **GNU**
RNAME operand coding requirements **RDU**
 root
 definition **ABDD**
 root anchor point (RAP) **ABDD, DIAG, GNU**
 root key
 definition **ABDD**
 root segment **CALL, DBA, CALL, HLPI**
 addition of **CALL**
 characteristics **DBA**
 definition **ABDD, GNU, HLPI**
 establishing position with **HLPI**
 example **DBA**
 HDAM sequence of **HLPI**
 in data base record **ABDD**
 key field in **ABDD, CALL, HLPI**
 qualification on, warning **HLPI**
 restriction **CALL**
 retrieval of **CALL**
 routine, creating ISQL EXTRACT DEFINES **RDU**
 routines

abnormal termination GNU
 general IRDU
 ASSGN IRDU
 DLBL IRDU
 EXEC statement IRDU
 JOB names IRDU
 PROC names IRDU
 TLBL IRDU
 RPG command (RQDLI) DIAG
 RPG II CALL, GNU
 *ENTRY PLIST CALL
 under CICS/VS CALL
 /INSERT statement in RPG II CALL
 "B" in Header Specification CALL
 application example GNU
 application program CALL
 batch program example CALL
 entry point CALL
 error processing example CALL
 examples of online calls CALL
 file-name CALL
 function names CALL
 indicator CALL
 job control for compile and link CALL
 link-editing of CALL
 online call CALL
 scheduling call CALL
 termination call CALL
 PCB mask CALL
 RQDLI format CALL
 termination of CALL
 Auto Report CALL, GNU
 batch program structure CALL, GNU
 example CALL
 checkpoint identification CALL
 communication with DL/I (see RQDLI command) CALL
 compile and link edit GNU
 continuation line (K-line) CALL
 DB (data base) file definition CALL
 DL/I application program for RPG II CALL
 DL/I batch program call (see DL/I application
 program for RPG II) CALL
 ELEM statement format CALL
 CHKPID option CALL
 FROM | INTO option CALL
 PCB option CALL
 SSA option CALL
 SSA-list option CALL
 entry to application program CALL
 error processing CALL
 example CALL
 standard error routine CALL
 indicator, translator use CALL
 job control GNU
 job control statements CALL
 UPSI CALL
 K-line (see continuation line) CALL
 language interface module CALL
 PCB mask CALL
 RQDLI commands CALL
 in an RPG II program (online) CALL
 for DB access CALL
 SETON LR CALL
 SSA specification in RPG-like format CALL
 ELIST command CALL
 QSSA statement CALL
 SSALIST option CALL
 USSA statement CALL
 standard data transfer CALL
 unresolved external reference CALL

statements for SSA specification CALL
 Translator CALL
 automatic *ENTRY PLIST generation CALL
 Translator output CALL, GNU
 UIB (user interface block) CALL
 scheduling call CALL
 ubiparm CALL
 user interface block (see UIB) CALL
 RPL, locating DIAG
 RQDLI DIAG
 commands GNU
 for DB access GNU
 format of GNU
 function names GNU
 in RPG II program CALL
 indicator GNU
 under CICS/VS GNU
 RTNAME operand coding requirements RDU
 SENFLD statement RDU
 VIRFLD statement RDU
 RTNAME operand, VIRFLD statement (PSBGEN) GNU
 rules GNU, LLC
 assignment of code values LLC
 for data base positioning after an HLPI command GNU
 for data base structures GNU
 for defining logical data bases GNU
 for defining physical data bases GNU
 for establishing logical relationships in parts
 data base LLC
 for names of DBDs, PSBs, and ACBs LLC
 intent scheduling DIAG
 parameter of SEGM statement LLC
 segment flag code DIAG
 segment processing DIAG
 RULES operand GNU
 LCHILD statement GNU
 of SEGM statement for a logical child GNU
 SEGM statement GNU
 RULES operand coding requirements RDU
 in LCHILD statements RDU
 logical child segments RDU
 in SEGM statements RDU
 HD data bases RDU
 HDAM data bases RDU
 HIDAM data bases RDU
 HISAM data bases RDU
 logical child segments RDU
 logical parent segments RDU
 physical parent segments in a logical relationship RDU
 SHISAM data bases RDU
 RULES parameter of DBDGEN, segment insertion CALL
 rules, data base positioning GNU
 rules, defining the DBA
 run and buffer statistics for online and MPS DBA
 RUN command RDU
 in ISQL EXTRACT DEFINES utility RDU
 run statistics DBA
 running DL/I programs in your system RDU

S

S type online trace request DIAG
 SA field in an FSE DIAG
 sample LLC
 application programs LLC
 Assembler language LLC
 COBOL LLC

control data base initialization **LLC**
 DBDs **LLC**
 for control data base **LLC**
 for parts data base **LLC**
 definition of control data base to VSAM **LLC**
 execution of LLC/CC in DL/I **LLC**
 initialization program for control data base (macro
 DLZNNICT) **LLC**
 macro
 DLZNN **LLC**
 DLZNNICT **LLC**
 PSBs **LLC**
 for control data base **LLC**
 for executing Assembler application **LLC**
 programs **LLC**
 sample application **ABBD, GNU**
 application analysis **ABBD**
 creating a data dictionary **ABBD**
 defining programs **ABBD**
 defining tasks **ABBD**
 naming convention **ABBD**
 CICS/VS-DL/I tables for DLZSAM60 **GNU**
 combining local views **ABBD**
 creating local views **ABBD**
 three step process **ABBD**
 customer data base **ABBD, GNU**
 description **GNU**
 customer history segment (STSCHIS), description **GNU**
 customer location segment (STSCLOC), description **GNU**
 customer name and address segment (STSCCST), description **GNU**
 customer status segment (STSCSTA), description **GNU**
 delete rule for
 logical child segment **GNU**
 logical parent segment **GNU**
 physical parent segment **GNU**
 DLZSAMCP, sample compression/expansion routine **GNU**
 insert rule for
 logical child segment **GNU**
 logical parent segment **GNU**
 physical parent segment **GNU**
 inventory data base **ABBD, GNU**
 description **GNU**
 inventory item segment (STPIITM), description **GNU**
 inventory location segment (STSILOC), description **GNU**
 job stream **GNU**
 naming conventions **GNU**
 objectives **ABBD**
 obtaining application requirements **ABBD**
 online application program **GNU**
 order item segment (STCCITM), description **GNU**
 phase 1 description **GNU**
 phase 2 description **GNU**
 phase 3 description **GNU**
 replace rule for
 logical child segment **GNU**
 logical parent segment **GNU**
 physical parent segment **GNU**
 substitute item segment (STCISUB), description **GNU**
 system view to physical view **ABBD**
 vendor name segment (STSIVND), description **GNU**
 sample application programs **GNU**
 sample basic PSBs **GNU**
 sample batch print program, DLZSAM50 **GNU**
 description **GNU**
 sample data requirements **DBA**
 sample data set definition **RDU**
 sample Gantt chart **GNU**
 sample jobstream, contents of **GNU**
 sample load program **DBA, HLPI**
 DLZSAM40 **GNU**
 description **GNU**
 example **GNU**
 sample panels **IRDU**
 sample print program, DLZSAM50 **GNU**
 description **GNU**
 sample programs **GNU**
 HLPI **GNU**
 sample project plan **DBA**
 data base implementation **GNU**
 sample PSBs **GNU**
 for phase 1 **GNU**
 for phase 2 **GNU**
 sample query routines **RDU**
 for Documentation Aid facility **RDU**
 sample recovery procedure flowchart **GNU**
 save area format **DIAG**
 save areas, register **DIAG**
 saving generated job stream output **IRDU**
 saving table output **IRDU**
 SBIF (subpool information table) **DIAG**
 scan function, IUG **DIAG**
 SCAN operand coding requirements **RDU**
 coding requirements **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 DATASET statement **GNU**
 scan utility **DIAG, RRG**
 SCD (system contents directory) **DIAG**
 get SCD call **DIAG**
 obtaining address of **DIAG**
 SCDDLIUP, as an initialization indicator **DIAG**
 SCHEDULE command **HLPI**
 function **HLPI**
 in programming example **HLPI**
 parentage **HLPI**
 position pointer **HLPI**
 segment selection **HLPI**
 status codes **HLPI**
 syntax **HLPI**
 SCHEDULE example **GNU**
 COBOL **GNU**
 PL/I **GNU**
 scheduling **DBA, HLPI**
 a PSB **GNU, HLPI**
 DL/I system **DBA**
 installation **DBA**
 logic **DBA**
 machine and programming system **DBA**
 request **DBA**
 system test **DBA**
 training **DBA**
 for short duration **GNU**
 scheduling call **CALL, GNU**
 checking the response to **CALL**
 scheduling command (HLPI) **GNU**
 format of **GNU**
 scheduling conflict or MAXTASK **DIAG**
 scheduling mechanism **GNU**
 intent scheduling **GNU**
 program isolation **GNU**
 selecting **GNU**
 scheduling MPS or non-MPS execution **GNU**
 SDBPHYCD **DIAG**
 search fields **DBA**
 secondary index **GNU**
 ACCESS statement **GNU**
 REF operand **GNU**
 SEGM operand **GNU**
 SEQFLD operand **GNU**

- SEQSEG operand GNU
- SEQVAL operand GNU
- SUPRTN operand GNU
- SUPVAL operand GNU
- accessing segments GNU
- adding a DBA
- creation of GNU
- deleting segments GNU
- design rules GNU
- index pointer segment GNU
- index source segment GNU
- index target segment GNU
- inserting segments GNU
- processing with GNU
- PROCSEQ operand, PCB statement (PSBGEN) GNU
- PSB coding GNU
- reorganization DBA
- replacing segments GNU
- retrieving segments GNU
- secondary processing sequence GNU
- segment types GNU
- when to use GNU
- secondary INDEX data bases RDU
 - coding examples RDU
 - control statements, format RDU
 - DATASET RDU
 - DBD RDU
 - DBDGEN RDU
 - END RDU
 - FIELD RDU
 - FINISH RDU
 - LCHILD RDU
 - SEGM RDU
 - control statements, summary RDU
 - how to define RDU
 - (see also secondary indexing, how to define in HDAM and HIDAM data base) RDU
 - input structure and rules RDU
- secondary indexes DBA, GNU, RDU
 - candidate ADBD
 - creating GNU
 - suppressing creation RDU
 - example RDU
- secondary indexing CALL, GNU, RDU
 - as direct access considerations ADBD
 - as structural considerations ADBD
 - characteristics of CALL
 - data base ADBD
 - definition ADBD
 - in HD data bases RDU
 - in HDAM data bases RDU
 - in HIDAM data bases RDU
 - index suppression CALL
 - indexed data CALL
 - processing sequence CALL
 - retrieving segments based on dependent's qualification ADBD
 - system-maintained data CALL
 - terms ADBD
 - pointer segment ADBD
 - source segment ADBD
 - target segment ADBD
 - user-maintained data CALL
 - uses ADBD, CALL
 - using a different key ADBD
 - with SSAs CALL
- secondary indexing exit routine interface DBA
- secondary indexing, how to define in HDAM and HIDAM data bases RDU
 - coding example RDU
 - control statements, summary RDU

- how to define RDU
 - (see also secondary INDEX data base) RDU
- segment types used RDU
 - index source segments RDU
 - index target segments RDU
- secondary indexing, space sequencing DBA
- secondary key
 - definition ADBD
- secondary processing sequence GNU
 - secondary index GNU
- security DBA
 - CICS/VS GNU
 - data DBA
- SEGLENGTH option GNU, HLPI
 - example of GNU
 - in COBOL program HLPI
 - in PL/I program HLPI
 - length of GNU
 - use with COBOL GNU
 - use with PL/I GNU
 - with fixed length segments, warning HLPI
 - with variable length segments HLPI
- SEGM operand GNU
 - coding requirements RDU
 - HD primary indexed access RDU
 - HD primary randomized access RDU
 - HD secondary indexed access RDU
 - primary indexed access GNU
 - primary randomized access GNU
 - secondary index, ACCESS statement GNU
- SEGM statement GNU
 - BYTES operand GNU
 - coding GNU
 - D operand, in logical DBD GNU
 - DATA operand, in logical DBD GNU
 - logical DBD GNU
 - SOURCE operand, in logical DBD GNU
 - COMPRTN operand GNU
 - format RDU
 - HD data bases RDU
 - HDAM data bases RDU
 - HIDAM data bases RDU
 - HISAM data bases RDU
 - HSAM data bases RDU
 - logical child segments RDU
 - LOGICAL data bases RDU
 - primary INDEX data bases RDU
 - secondary INDEX data bases RDU
 - SHISAM data bases RDU
 - SHSAM data bases RDU
 - virtual logical child segments RDU
 - logical child GNU
 - BYTES operand GNU
 - NAME operand GNU
 - PARENT operand GNU
 - POINTER operand GNU
 - RULES operand GNU
 - logical parent GNU
 - NAME operand GNU
 - PARENT operand GNU
 - POINTER operand GNU
 - SOURCE operand GNU
 - NAME operand GNU
 - PARENT operand GNU
 - POINTER operand GNU
 - RULES operand GNU
- segment ADBD, CALL, GNU, HLPI
 - accessing in a logical DBD GNU
 - accessing with secondary index GNU
 - adding new types to a data base structure GNU

code GNU
compared to physical record **ABDD**
concatenated GNU, **HLPI**
 variable length **HLPI**
correspondence to data aggregate **ABDD**
definition **CALL**, GNU, **HLPI**
dependent **ABDD**, GNU
 definition **ABDD**
edit/compression exit GNU
expansion, dynamic **CALL**, GNU
fixed length **HLPI**
format GNU
grouping GNU
I/O area GNU
in data base record **HLPI**
index pointer GNU
index source GNU
index target GNU
lengths, maximum GNU
logical child GNU
logical parent GNU
maximum length **HLPI**
multiple occurrences **CALL**
object GNU
option GNU
physical parent GNU
root **ABDD**, GNU
 definition **ABDD**
 key field in **ABDD**
selection GNU
sensitivity GNU
sort control field GNU
twin GNU
types
 and their relationships in a hierarchical
 data structure GNU
 associated with a secondary index GNU
 involved in logical relationships GNU
 numbered in hierarchical sequence GNU
variable length GNU, **HLPI**
segment compression routines, where defined (see
COMPRTN operand) **RDU**
segment contents
 defining **DBA**
 specifying **DBA**
segment edit/compression **ABDD**, **DBA**
 conversion to use of **DBA**
 exit, considerations for GNU
 routine **ABDD**
 functions **DBA**
 interface **DBA**
 segment compression table **DBA**
 segment expansion **DBA**
segment editing GNU
segment expansion **DBA**
 dynamic **DBA**
segment field name in qualified SSA **CALL**
segment flag **DIAG**
 codes **DIAG**
 DBDGEN assembler listing **DIAG**
segment format
 HSAM or **HISAM** **DIAG**
 simple **HSAM** or simple **HISAM** **DIAG**
segment format, logical child GNU
segment hierarchy level indicator GNU
segment I/O area **HLPI**
segment insertion **CALL**
 example of **CALL**
segment intent list **DIAG**
segment length GNU, **HLPI**
 defining **DBA**
segment level
 enqueueing **DBA**
 indicator in PCB mask **CALL**
segment name **CALL**, GNU
 feedback area GNU
 feedback in PCB mask **CALL**
 in SSA **CALL**, GNU
segment name, defining the **DBA**
segment occurrence **CALL**, **DBA**, **CALL**
 defined **DBA**
 example **DBA**
 illustrated **DBA**
 number **CALL**, **DBA**
 retrieval of last **CALL**
 sequencing **DBA**
 sequential processing **DBA**
 control by application program **DBA**
 control by field value **DBA**
 control by order of creation **DBA**
 sparse sequencing **DBA**
SEGMENT operand coding requirements **RDU**
SEGMENT option **HLPI**
 detailed description **HLPI**
 introduction **HLPI**
segment organization, defining **DBA**
 left-to-right ordering of children **DBA**
 parent/child relationships **DBA**
segment prefixes **DBA**
 illustration of **DIAG**
 updating **DBA**
segment processing options, defining **DBA**
 process control for fields **DBA**
 process controls for segments **DBA**
segment processing rules **DIAG**
segment qualification
 field level sensitivity GNU
 statement **CALL**
 in SSA **CALL**
 parts of **CALL**
segment search argument (see SSA) **CALL**, **DIAG**, GNU
 format **DIAG**
segment selection **HLPI**
 boolean operators **HLPI**
 example **HLPI**
 in **WHERE** option **HLPI**
 example **HLPI**
segment selection, qualified GNU
segment sensitivity **DBA**
segment sequence checking, HD unload **DIAG**
segment size **HLPI**
 changing **DBA**
 in **INSERT** command **HLPI**
 in **REPLACE** command **HLPI**
 in variable length segment **HLPI**
 minimum valid value **HLPI**
 starting position in variable length segment **HLPI**
segment table **DIAG**
 DBDGEN assembler listing **DIAG**
segment types **CALL**, **DBA**, GNU
 adding **DBA**
 using reorganization utilities **DBA**
 using your own program **DBA**
 where added **DBA**
 without unloading or reloading **DBA**
 and occurrences **DBA**
 changing the sequence of **DBA**
 child **DBA**
 definition **ABDD**, **DBA**

- deleting segment types **DBA**
- example **DBA**
- illustrated **DBA**
- left-to-right ordering of **DBA**
 - with sequential connections **DBA**
- logical child **DBA, GNU**
- logical parent **GNU**
- maximum number **CALL, DBA**
 - in data base record **HLPI**
- parent **DBA**
- physical parent **GNU**
- root **DBA**
- sequence **DBA**
- target **DBA**
- using reorganization utilities **DBA**
- segments **DBA**
 - and fields **DBA**
 - changing the position of data **DBA**
 - characteristics **DBA**
 - child **DBA**
 - combining **DBA**
 - compression table **DBA**
 - concatenated **DBA**
 - connecting child and parent **DBA**
 - direct pointer method **DBA**
 - example **DBA**
 - layout of record
 - direct pointer method **DBA**
 - sequential method **DBA**
 - sequential method **DBA**
 - contents, defining **DBA**
 - defining **DBA**
 - field location within **DBA**
 - dependent **DBA**
 - example **DBA**
 - inserting **DBA**
 - lengths **DBA**
 - defining **DBA**
 - logical child, defining **DBA**
 - parent operand **DBA**
 - maximum size **DBA**
 - name, defining **DBA**
 - occurrence **DBA**
 - parent **DBA**
 - prefix **DBA**
 - relationship between **DBA**
 - relative byte address (RBA) **DBA**
 - root **DBA**
 - selection **DBA**
 - sensitivity **DBA**
 - specifying **DBA**
 - twin **DBA**
 - type **DBA**
 - variable length, adding or converting to **DBA**
- SELECT** statement **RDU**
 - in ISQL EXTRACT DEFINES utility **RDU**
- selection of segments **HLPI**
- selective restore **RRG**
- selective unload **RDU**
- SENFLD macro, specifying data type **GNU**
- SENFLD statement **GNU**
 - BYTES operand **GNU**
 - format **RDU**
 - NAME operand **GNU**
 - REPLACE operand **GNU**
 - RTNAME operand **GNU**
 - START operand **GNU**
 - TYPE operand **GNU**
- SENSEG statement (PSBGEN) **GNU**
 - format **RDU**

- NAME operand **GNU**
- PARENT operand **GNU**
- PROCOPT operand **GNU**
- sensitive fields **DBA**
- sensitive segments **GNU**
 - number of in PCB mask **CALL**
- sensitivity **GNU**
 - definition of **CALL**
 - field **CALL**
 - level **GNU**
 - segment **GNU**
- SEQ parameter, NAME operand of FIELD statement **GNU**
- SEQFLD operand **GNU**
 - coding requirements **RDU**
 - HD primary indexed access **RDU**
 - HD primary randomized access **RDU**
 - HD secondary indexed access **RDU**
 - primary indexed access **GNU**
 - primary randomized access **GNU**
 - secondary index, ACCESS statement **GNU**
- SEQSEG operand
 - coding requirements **RDU**
 - secondary index, ACCESS statement **GNU**
- sequence checking, segment **DIAG**
- sequence field **DBA, GNU**
 - (see key field) **CALL, HLPI**
 - and access paths **GNU**
 - in a data base record **GNU**
- sequence
 - hierarchical **HLPI**
 - segment **DBA**
 - specification of **DBA**
- sequencing **DBA**
 - by field value **DBA**
 - for HD **DBA**
 - for non-HD **DBA**
 - by order of creation **DBA**
 - direct pointer **DBA**
 - indexed **DBA**
 - records, controlling **DBA**
 - sequential **DBA**
 - under application control **DBA**
- sequential access **ADBD**
 - program **GNU**
- sequential order
 - records stored in **ADBD**
- sequential organization **ADBD, DIAG**
 - advantages **ADBD**
 - definition **ADBD**
 - disadvantages **ADBD**
 - HISAM **ADBD**
 - HSAM **ADBD**
 - simple HISAM **ADBD**
 - simple HSAM **ADBD**
- sequential organization access methods **DBA**
 - hierarchical indexed sequential access method (HISAM) **DIAG**
 - hierarchical sequential access method (HSAM) **DIAG**
 - simple hierarchical indexed sequential access method (simple HISAM) **DIAG**
 - simple hierarchical sequential access method (simple HSAM) **DIAG**
- sequential processing **DBA**
 - using secondary indexing **DBA**
- sequential retrieval **HLPI**
 - sequential retrieval, using GN call **CALL**
- sequential search direct access **DBA**
- SEQVAL operand **GNU**
 - primary randomized access **GNU**
 - secondary index, ACCESS statement **GNU**
- SEQVAL operand coding requirements **RDU**
 - HD primary randomized access **RDU**

HD secondary indexed access **RDU**
 set exit abnormal **HLPI**
 share option 1, VSAM **GNU**
 share option 2, VSAM **GNU**
 share option 3, VSAM **GNU**
 share option 4, VSAM **GNU**
SHISAM ADBD
 and HISAM indexes **DBA**
SHISAM data bases RDU
 coding example **RDU**
 control statements, format **RDU**
 DATASET RDU
 DBD RDU
 DBDGEN RDU
 END RDU
 FIELD RDU
 FINISH RDU
 SEGM RDU
 control statements, summary **RDU**
 how to define **RDU**
 input structure and rules **RDU**
 reorganizing **DBA**
SHSAM ADBD
 data bases **RDU**
 coding example **RDU**
 control statements **RDU**
 DATASET RDU
 DBD RDU
 DBDGEN RDU
 END RDU
 FIELD RDU
 FINISH RDU
 format **RDU**
 SEGM RDU
 summary **RDU**
 how to define **RDU**
 input structure and rules **RDU**
 shut-down of CICS/VS, caution **GNU**
 simple hierarchical indexed sequential access
 method (simple HISAM) **GNU**
 simple hierarchical sequential access method (simple HSAM) **GNU**
 simple HISAM **ADBD**
 simple HISAM (simple hierarchical indexed sequential
 access method) **DBA, GNU**
 presort for loading **HLPI**
 simple HSAM **ADBD, HLPI**
 access method **DBA**
 key field not necessary **HLPI**
 presort for loading **HLPI**
 simple HSAM or simple HISAM segment format **DIAG**
 single arrow link
 between keys in color **ADBD**
 with concatenated keys **ADBD**
 single occurrence of data aggregate **ADBD**
 single positioning **CALL, HLPI**
 example of **CALL**
SIT (system initialization table) CICS/VS **DBA, GNU, RDU, RRG**
 contents **GNU**
 size of a logical unit of work **RRG**
SIZE parameter in EXEC statement **CALL**
 skeletons
 for ACT **DIAG**
 for DBD **DIAG**
 for PSB **DIAG**
 skip, sequential processing **CALL**
SLC (storage layout control table) **DBA, RDU**
 control statements, format **RDU**
 DLZSLC TYPE=ENTRY RDU
 DLZSLC TYPE=FINAL RDU
 DLZSLC TYPE=INITIAL RDU
 control statements, summary **RDU**
 how to define **RDU**
 (see also SLC operand) **RDU**
 how to generate **RDU**
 input structure and rules **RDU**
 operand coding requirements **RDU**
 parameter, DLZACT macro **GNU**
 sort control field **DBA**
 segment **GNU**
 sort table, IUG created **DIAG**
 SORT tables **IRDU**
 sorting segments into hierarchical sequence **GNU**
 SOURCE operand **GNU**
 SEGM statement
 for a logical parent **GNU**
 in logical DBD **GNU**
 source segment **ADBD**
 space allocation **DBA**
 data base **DBA**
 procedure **DBA**
 reason for change **DBA**
 space management search algorithm **DBA**
 space management utilities IUP **DBA**
 HD pointer checker utility **DBA**
 HD unload utility **DBA**
 HDAM physical block reload utility **DBA**
 pointer checker **DBA**
 function **DBA**
 purpose **DBA**
 space manager, DL/I **DBA**
 space sequencing
 based on segment occurrence **DBA**
 based on sequence field value **DBA**
 sparse sequencing **DBA**
 based on inspection by user routine **DBA**
 based on sequence field value **DBA**
 sparse sequencing based on inspection by user routine **DBA**
 sparse sequencing using secondary indexing **DBA**
 Special call statements **DIAG**
 specifying **HLPI**
 the checkpoint ID **HLPI**
 the PCB **HLPI**
 the PSB **HLPI**
 specifying a data base resident on another system **GNU**
 specifying buffer pool control options **GNU**
 specifying the end of the DL/I application control table **GNU**
 split screen
 precautions **IRDU**
 SQL/DS **RDU**
 extracting DL/I data to **RDU**
 SQL/DS tables **DIAG, GNU, RDU**
 accessing data in SQL/DS tables **DBA**
 creating for ISQL EXTRACT DEFINES utility **RDU**
 DBD information in **DBA**
 DBDACCESSDATA **DBA, RDU**
 DBDBASICDATA **DBA, RDU**
 DBDFIELDDATA **DBA, RDU**
 DBDLCHILDDATA **DBA, RDU**
 DBDSEGMENTDATA **DBA, RDU**
 displaying data **RDU**
 in ISQL EXTRACT DEFINES utility **RDU**
 documentation aid **GNU**
 tables **DBA**
 in Documentation Aid facility **DBA, RDU**
 accessing data **RDU**
 loading data into **RDU**
 in ISQL EXTRACT DEFINES utility **RDU**
 PSBBASICDATA **DBA, RDU**
 PSBFIELDDATA **DBA, RDU**
 PSBPCBDATA **DBA, RDU**

PSBSEGMENTDATA DBA, RDU
 storing data base definitions **DBA**
 storing DBD information in **RDU**
 storing PSB information in **RDU**
SRCH operand coding requirements **RDU**
SSA DIAG
SSA (segment search argument) **CALL**
 command code **CALL**
 field level sensitivity **CALL**
 for DLET call **CALL**
 for GN (GHN) call **CALL**
 for GNP (GHNP) call **CALL**
 for GU (GHU) call **CALL**
 for ISRT (add) call **CALL**
 for ISRT (load) call **CALL**
 for REPL call **CALL**
 format **DIAG**
 with command code **CALL**
 implied **CALL**
 length
 restriction **DIAG**
 Boolean operators and **CALL**
 multiple positioning **CALL**
 omission of **CALL**
 retrieval of segment **CALL**
RPG II GNU
 basic function **GNU**
 command codes **GNU**
 comparative value **GNU**
 number per hierarchical level **GNU**
 qualification statement **GNU**
 relational operator **GNU**
 segment name **GNU**
 specification in RPG-like format **GNU**
 statements for **GNU**
 segment name in **CALL**
 when used **CALL**
SSA, elements of CALL
 command code **CALL**
 segment name **CALL**
 segment qualification statement **CALL**
SSALIST option **GNU**
 standard data transfer unresolved external
 references (RPG II) **CALL**
 standard RPG II data transfer **GNU**
start DBD DBA
START operand coding requirements **RDU**
 in **FIELD** statement **RDU**
 /CK system-related fields **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**
 HISAM data bases **RDU**
 HSAM data bases **RDU**
 SHISAM data bases **RDU**
 SHSAM data bases **RDU**
 in **SENFLD** statement **RDU**
 in **VIRFLD** statement **RDU**
START operand, **FIELD** statement **GNU**
START operand, **VIRFLD** statement (PSBGEN) **GNU**
 start-stop time, device **GNU**
START, DLZTRCAL macro **DIAG**
START=AUTO RDU
 MPS restart facility **RDU**
 statements
BUILD DIAG
 job control for **ACTGEN DIAG**
 job control for **DLZDLTXX DIAG**
 UDR for data set recovery utility **DIAG**
 statistical output, buffer **DBA**
 statistics
 buffer pool for batch **DBA**
 examples of use **DBA**
 run and buffer for online and MPS **DBA**
 status codes **CALL, GNU, HLPI**
 after **CHKP** call **CALL**
 after **DLET** call **CALL**
 after **GN (GHN)** call **CALL**
 after **GNP (GHNP)** call **CALL**
 after **GU (GHU)** call **CALL**
 after **ISRT (add)** call **CALL**
 after **ISRT (load)** call **CALL**
 after **REPL** call **CALL**
AM HLPI
 as debugging aids **HLPI**
 description **HLPI**
 example for testing **GNU**
 for **RPG II GNU**
 in **PCB** mask **CALL**
 introduction **HLPI**
 list of **DL/I CALL**
 online trace **DIAG**
 summary of **CALL, GNU**
 chart **HLPI**
 with **GSCD** call **DIAG**
 status statement format **DIAG**
STCCITM (order item segment), description **GNU**
STCISUB (substitute item segment), description **GNU**
 steps in data base design **GNU**
STOP and **STRT** call example **DBA**
STOP call **DBA**
 stop **DBD DBA**
STOP RUN (COBOL) GNU
 restriction of **CALL**
STOP, DLZTRCAL macro **DIAG**
STOPKEY operand of **DLZTRACE DIAG**
 storage
 assignment, real **DBA**
 layout control option **DBA**
 space, scanning for **DBA**
 storage considerations
MPS GNU
 with program isolation **CALL**
 storage errors, temporary **DIAG**
CICS/VS Command Interpreter DIAG
MPS restart DIAG
TSQ entry table **DIAG**
 with **MPS Restart DIAG**
 storage layout control table (see **SLC** table) **RDU**
STPCORD (customer order segment), description **GNU**
STPIITM (inventory item segment), description **GNU**
STRT and **STOP** call example **DBA**
STRT call **DBA**
STRTKEY operand of **DLZTRACE DIAG**
 structural elements **DBA**
 structural organizations **DBA**
 structure changes, hierarchical **DBA**
 structure definitions, details for **DBA**
 structure of a batch **RPG II** application program **GNU**
 structure of a physical data base (see physical data structure) **GNU**
Structured Query Language/Data System(SQL/DS) RDU
 structures
 data base **DBA**
 defining extended **DBA**
 extended **DBA**
 features of extended **DBA**
 hierarchical **DBA**
 system calls **DBA**

STSCCST (customer name and address segment), description **GNU**
 STSCHIS (customer history segment), description **GNU**
 STSCLOC (customer location segment), description **GNU**
 STSCSTA (customer status segment), description **GNU**
 STSILOC (inventory location segment), description **GNU**
 STSIVND (vendor name segment), description **GNU**
 STXIT
 AB DIAG, GNU
 abend routine **DIAG**
 abnormal linkage **CALL**
 PC DIAG, GNU
 restriction **GNU**
 subfields **DBA**
 SUBMIT command **RDU**
 in ISQL EXTRACT DEFINES utility **RDU**
 subpools **GNU**
 allocation, default **DBA**
 information table (SBIF) **DIAG**
 SUBSEQ operand coding requirements **RDU**
 subsequence fields (see SUBSEQ operand) **DBA, RDU**
 substitute item segment (STCISUB), description **GNU**
 SUFFIX operand coding requirements **RDU**
 SUFFIX parameter, DLZACT macro **GNU**
 summary of control statements **RDU**
 ACTs RDU
 basic PSBs **RDU**
 field level sensitivity **RDU**
 HD DBDs RDU
 HDAM DBDs RDU
 HIDAM DBDs RDU
 HISAM DBDs RDU
 HSAM DBDs RDU
 LOGICAL DBDs RDU
 logical relationships in physical DBDs **RDU**
 logical child segments **RDU**
 logical parent segments **RDU**
 virtual logical child segments **RDU**
 primary INDEX DBDs **RDU**
 secondary INDEX DBDs **RDU**
 secondary indexing in physical DBDs **RDU**
 index source segments **RDU**
 index target segments **RDU**
 SHISAM data bases RDU
 SHSAM data bases RDU
 SLC tables RDU
 supervisor XECB table **DIAG**
 suppressing creation of secondary indexes example **RDU**
 SUPRTN operand
 coding requirements **RDU**
 secondary index, ACCESS statement **GNU**
 SUPVAL operand
 coding requirements **RDU**
 secondary index, ACCESS statement **GNU**
 syncpoint record, CICS/VS **GNU**
 synchronization **GNU**
 checkpoint **GNU**
 synchronization point, CICS/VS **HLPI**
 using **RRG**
 online transactions **RRG**
 synchronous logging **RRG**
 synonyms **ADBD, GNU**
 data elements **GNU**
 HD GNU
 syntax **GNU, HLPI**
 description **HLPI**
 introduction **HLPI**
 notations (RPG II) **GNU**
 of HLPI commands **GNU**
 of the command language **HLPI**
 summary **HLPI**
 chart **HLPI**
 SYSGEN parameters for DL/I-CICS/VS **DIAG**
 SYSID operand coding requirements **RDU**
 SYSLSLST option of DLZTRACE OUTPUT operand **DIAG**
 system **GNU**
 DL/I batch **GNU**
 generation, CICS/VS **GNU**
 initialization table (SIT), CICS/VS **GNU**
 initialization, online **GNU**
 installation, user responsibilities **GNU**
 journal, CICS/VS **GNU**
 table preparation, CICS/VS **GNU**
 system calls **CALL, DBA, DIAG**
 controlling use of **DBA**
 examples **DBA**
 CMXT DBA
 STOP DBA
 STRT DBA
 TSTP DBA
 TSTR DBA
 general structure **DBA**
 in DLZMDLI **DIAG**
 invalid password **DBA**
 parameter requirements **DBA**
 scheduling **DBA**
 special scheduling **DBA**
 format **DBA**
 TCA status codes **DIAG**
 TSTP (trace stop) **DIAG**
 TSTR (trace start) **DIAG**
 system configuration **LLC**
 system contents directory (SCD) **DIAG**
 system control facility **DBA**
 system control flow **DBA**
 functions **DBA**
 initializing the batch system **DBA**
 initializing the data base system **DBA**
 illustrated **DBA**
 system execution **DBA**
 system implementation **DBA**
 system initialization table (SIT), CICS/VS **DBA, RDU, RRG**
 system journal, CICS/VS **DBA**
 system log
 (see log file) **CALL**
 DL/I **DBA**
 tapes **DBA**
 system operation **DBA**
 system password address **DIAG**
 system programming **DBA**
 system requirements **DBA**
 system initialization **DBA**
 MPS DBA
 system storage, DL/I MPS **DBA**
 system view **ADBD, DBA**
 combining local views **ADBD**
 example **ADBD**
 procedure **ADBD**
 defining implementation requirements **ADBD**
 definition **ADBD, DBA**
 eliminating redundancies **ADBD**
 generating **ADBD**
 operations **ADBD**
 identifying keys **ADBD**
 concatenated **ADBD**
 primary **ADBD**
 secondary **ADBD**
 intersecting attributes **ADBD**
 intersection data **ADBD**
 isolated attributes **ADBD**
 local views as input **ADBD**

- mapping between keys **ABDD**
- procedure **ABDD**
- relationships **ABDD**
 - many-to-many **ABDD**
 - many-to-one **ABDD**
 - one-to-many **ABDD**
 - one-to-one **ABDD**
- removing undesirable associations **ABDD**
- system-maintained data **CALL**
- system-maintained fields **GNU**
- systems, programming **LLC**

T

- T type online trace request **DIAG**
- table
 - ACT** **DIAG**
 - DBD** **DIAG**
 - list of ACT definitions **IRDU**
 - list of DBD definitions **IRDU**
 - list of PSB definitions **IRDU**
 - migrated ELIAS **DIAG**
 - PSB** **DIAG**
- table data, user
 - CONTROL** **IRDU**
 - PART** **IRDU**
 - SORT** **IRDU**
- table output **IRDU**
 - saving **IRDU**
- tables
 - CICS/VS** **RDU**
 - DL/I** **DBA, RDU**
 - converting from IPF format to ISPF format **IRDU**
 - IUG** created **DIAG**
 - SQL/DS** **DBA, RDU**
 - in Documentation Aid facility **RDU**
- target segment **ABDD**
- target table, creating **RDU**
- task termination, **DL/I** abnormal **DBA**
- task-id field in an **FSE** **DIAG**
- tasks in data base design **GNU**
- tasks, controlling the number of **CICS/VS** and **DL/I** **GNU**
- TCA** field, flag bit values **DIAG**
- TCA** status codes, system calls **DIAG**
- TCADLII** field, flag bit values **DIAG**
- TCLASS** parameter, **CICS/VS** **GNU**
- technique and suggestions for writing programs **HLPI**
- temporary storage queue (**TST**), **CICS/VS** **DBA**
 - purging of **DBA**
- temporary storage table (**TST**), **CICS/VS** **GNU, RDU**
 - for **MPS** restart **GNU**
- TERM** (release a **PSB**) **GNU**
- TERM** (termination) call **CALL, DBA**
- TERMINATE** command **HLPI**
 - function **HLPI**
 - in programming example **HLPI**
 - parentage **HLPI**
 - position pointer **HLPI**
 - segment selection **HLPI**
 - status codes **HLPI**
 - syntax **HLPI**
- TERMINATE** example **GNU**
- COBOL** **GNU**
 - PL/I** **GNU**
- terminating **HLPI**
 - a **PSB** **GNU, HLPI**
 - an online program **GNU**
 - COBOL** **GNU**

- PL/I** **GNU**
 - command (**HLPI**) **GNU**
 - format of **GNU**
 - your program **GNU, HLPI**
 - batch and **MPS** batch **HLPI**
 - online **HLPI**
- termination
 - call **GNU**
 - in batch, abnormal **DBA**
 - in **CICS/VS**, abnormal **DBA**
 - in **MPS** partition, abnormal **DBA**
 - messages, abnormal **HLPI**
 - of application program **CALL**
 - online **CALL**
 - checking the response to **CALL**
 - routines, **DL/I** abnormal **DBA**
 - RPG II** **GNU**
- test data base **DBA**
 - generating a **DBA**
- test program **DLZDLTXX** **DIAG**
 - sample job control statements **DIAG**
 - suggestions for using **DIAG**
- test requirements **DBA**
- testing application programs **DBA**
- testing for the status code **GNU**
- three step process **ABDD**
 - grouping data elements with keys **ABDD**
- introduction **ABDD**
- isolating duplicate values **ABDD**
- isolating repeating data elements **ABDD**
- using **ABDD**

- TI** control statement
 - example **DIAG**
 - format **DIAG**
- TLBL** parameters, entering **IRDU**
- TO** control statement
 - examples **DIAG**
 - format **DIAG**
- tools
 - for improving performance **DBA**
 - data base design **ABDD**
- TRACE** **DIAG**
- trace entry
 - format (**DLZTRACE**) **DIAG**
 - header format (**DLZTRACE**) **DIAG**
 - online trace **DIAG**
 - selection for printing **DIAG**
- trace entry **DSECTs**
 - BHINTF** Type 1
 - TRBHF**FROM **DIAG**
 - TRBHT0** **DIAG**
 - CURRPOS**
 - TRCURPOS** **DIAG**
 - TRLEVEL** **DIAG**
 - TRSDB** **DIAG**
 - INDEXTRC**
 - TRINDEX** **DIAG**
 - MODTRACE**
 - TRMODTRC** **DIAG**
 - ONLINEBH**
 - TRONLINE** **DIAG**
 - PITRACE**
 - TRPITRC** **DIAG**
 - RETRIEVE**
 - TRETRV** **DIAG**
 - USERCALL** Type 1
 - TRUSRCL1** **DIAG**
 - USERCALL** Type 2
 - TRUSRCL** **DIAG**
 - VSAMINTF**

TRVSAMIF **DIAG**
 trace facilities **DBA**
 online **DIAG**
 trace types **DBA**
 trace information, online trace **DIAG**
 trace invocation macro (DLZTRCAL) for action modules **DIAG**
 TRACE operand, parameter statement **GNU**
 trace output
 batch environment **DIAG**
 online environment **DIAG**
 TRACE parameter **RDU**
 trace point parameter list **DIAG**
 trace points
 BHINTF Type 1
 DLZCPY10 (FLS Copy) **DIAG**
 DLZDDLE0 (load/insert) **DIAG**
 DLZDHDS0 (space management) **DIAG**
 DLZDL00 (delete/replace) **DIAG**
 DLZDLR00 (retrieve) **DIAG**
 DLZDXMT0 (index maintenance) **DIAG**
 CURRPOS
 DLZDLA00 (call analyzer) **DIAG**
 INDEXTRC
 DLZDXMT0 (index maintenance) **DIAG**
 MODTRACE
 DLZCPY10 (FLS copy) **DIAG**
 DLZDBH00 (buffer handler) **DIAG**
 DLZDDLE0 (load/insert) **DIAG**
 DLZDHDS0 (space management) **DIAG**
 DLZDLA00 (call analyzer) **DIAG**
 DLZDL00 (delete/replace) **DIAG**
 DLZDLR00 (retrieve) **DIAG**
 DLZDXMT0 (index maintenance) **DIAG**
 in buffer handler before VSAM call **DIAG**
 ONLINEBH
 AFTW routine **DIAG**
 ISSWAIT routine **DIAG**
 PSEUDINT routine **DIAG**
 PITRACE
 DLZQUEF0 **DIAG**
 DLZQUEF0 (queueing facility) **DIAG**
 in QWAIT **DIAG**
 RETRIEVE
 DLZEODC **DIAG**
 DLZGER **DIAG**
 DLZLTW **DIAG**
 DLZSSA **DIAG**
 DLZTAG **DIAG**
 USERCALL Type 1
 DLZDLA00 (call analyzer) **DIAG**
 USERCALL Type 2
 DLZDLA00 (call analyzer) **DIAG**
 VSAMINTF
 DLZDBH00 (buffer handler) **DIAG**
 trace print function, IUG problem determination **DIAG**
 trace print utility **DBA, DIAG**
 control statements **IRDU**
 description **IRDU**
 illustration **IRDU**
 input/output **IRDU**
 job control statement **DIAG**
 examples **DIAG**
 parameters **IRDU**
 trace start **DBA**
 trace stop **DBA**
 TRACEHDR **DIAG**
 tracing control flow by register save area **DIAG**
 tracing in the batch environment (DLZTRACE) **DIAG**
 sample DLZTRACE job **DIAG**
 trace output **DIAG**

what information to trace **DIAG**
 which calls to trace **DIAG**
 tracing in the online environment **DIAG**
 (See also online trace facilities)
 debugging **DIAG**
 defining the destination control table (DCT)
 example **DIAG**
 defining the extrapolation data set **DIAG**
 general trace **DIAG**
 opening the CICS/VS device **DIAG**
 trace output **DIAG**
 tracing control **DIAG**
 tracing, parameter statement **CALL**
 TRACSIZ operand of DLZTRACE **DIAG**
 traditional recovery approach **GNU**
 training, personnel **DBA**
 transaction **GNU**
 transaction/data element matrix **GNU**
 transaction, data base **GNU**
 transactions in online **CALL**
 translate, compile, and link edit **GNU**
 translation **HLPI**
 and compilation (RPG II) **GNU**
 translator **HLPI**
 CICS/VS **GNU**
 RPG II **GNU**
 TRCECON operand of DLZTRACE **DIAG**
 tree structure **ADBD**
 (See also hierarchical structure)
 definition **ADBD**
 tree structure (see product tree structure) **LLC**
 trigger **GNU, HLPI**
 definition **HLPI**
 description **HLPI**
 example **HLPI**
 EXECUTE DLI **GNU**
 HLPI use **GNU**
 in syntax **HLPI**
 reserved keywords **HLPI**
 triggers, CICS/VS translator use of **GNU**
 truncation **HLPI**
 key feedback length **HLPI**
 TSQ entry table **DIAG**
 access error **DIAG**
 format **DIAG**
 unavailable **DIAG**
 TST (temporary storage table) **DBA, RDU**
 CICS/VS **GNU**
 for MPS restart **GNU**
 TSTP (trace stop) system call **DBA, DIAG**
 and TSTR call example **DBA**
 TSTR (trace start) system call **DBA, DIAG**
 and TSTP call example **DBA**
 twin segment **DBA**
 definition **DBA, GNU**
 illustrated **DBA**
 two logically related data bases, CUSTOMER
 and INVENTORY **GNU**
 TYPE operand **GNU**
 FIELD statement **GNU**
 PCB statement **GNU**
 VIRFLD statement (PSBGEN) **GNU**
 TYPE operand coding requirements **RDU**
 in BUFFER type DLZACT statement **RDU**
 in CONFIG type DLZACT statement **RDU**
 in ENTRY type DLZSLC statement **RDU**
 in FIELD statement **RDU**
 HD data bases **RDU**
 HDAM data bases **RDU**
 HIDAM data bases **RDU**

- HISAM data bases **RDU**
- HSAM data bases **RDU**
- primary INDEX data bases **RDU**
- secondary INDEX data bases **RDU**
- SHISAM data bases **RDU**
- SHSAM data bases **RDU**
- in FINAL type DLZACT statement **RDU**
- in FINAL type DLZSLC statement **RDU**
- in INITIAL type DLZACT statement **RDU**
- in INITIAL type DLZSLC statement **RDU**
- in PCB statement **RDU**
- in PROGRAM type DLZACT statement **RDU**
- in RPSB type DLZACT statement **RDU**
- in SENFLD statement **RDU**
- in VIRFLD statement **RDU**
- types of application programs **GNU**
- types of primary access **GNU**
 - indexed access **GNU**
 - randomized access **GNU**
- TYPETRC operand of DLZTRACE **DIAG**
- typical DL/I real storage requirements **GIM**

U

- UDR parameter information **DIAG**
- UDR statement **DIAG**
- UIB (user interface block) **CALL**
 - uibparm **CALL**
- ULU parameter information **DIAG**
- uncontrolled BPC abend **DIAG**
- undesirable associations
 - removing **ADBD**
- unidirectional logical relationship **GNU**
 - (see LCHILD statement) **RDU**
 - defining a **DBA**
 - example of **DBA**
- unidirectional relationships **DBA**
 - defining the rules for **DBA**
 - LCHILD, specification of **DBA**
 - versus bidirectional relationships **DBA**
- unload program, using your own **DBA**
- unload sequence checking **DIAG**
- unload/reload process **RRG**
 - copying data bases **RRG**
 - Data Base Prefix Resolution utility **RRG**
 - Data Base Prefix Update utility **RRG**
 - Data Base Preorganization utility **RRG**
 - Data Base Scan utility **RRG**
 - HD Reorganization Reload utility **RRG**
 - logical relationships **RRG**
 - logically related data bases **RRG**
 - related utilities **RRG**
 - uses of **RRG**
- unloading data bases **RRG**
- unqualified segment selection **HLPI**
- unqualified SSA **CALL**
- unresolved external references **DIAG**
- update call (see REPL call) **CALL**
- update function **GNU**
 - modes of **GNU**
 - rules for **GNU**
- update intent **DBA**
- update mode **LLC**
 - description **LLC**
 - details **LLC**
 - prerequisites **LLC**
 - if initial generation not done properly **LLC**
 - processing **LLC**

- use **LLC**
- updating **GNU, HLPI**
 - a data base **GNU**
 - a segment, example of **CALL**
- UPSI (user program switch indicator) **GNU, HLPI, RRG**
- format of **GNU**
- RPG II **GNU**
- setup for logging **RRG**
- settings **CALL, GNU, RDU**
 - for batch DL/I **DIAG, GNU, RDU**
 - log **CALL**
 - MPS **DIAG, GNU, RDU**
 - online **DIAG, GNU, RDU**
 - storage dump **CALL**
 - STXIT processing **CALL**
- user completion entry **DIAG**
- user error aids
 - batch **DIAG**
 - MPS batch **DIAG**
 - online **DIAG**
- user exit
 - description **DIAG**
 - EXIT 1 **DIAG**
 - EXIT 2 **DIAG**
 - EXIT 3 **DIAG**
 - interface **DIAG**
 - parameter list **DIAG**
 - sample routines (DLZTXIT0) **DIAG**
 - trace point parameter list **DIAG**
- user field exit routine **CALL, DBA, CALL**
- user interface block (see UIB) **CALL**
- user load program **IRDU**
- user parameter list, finding **DIAG**
- user program switch indicator (see UPSI byte) **CALL, HLPI**
- user responsibilities **GNU**
 - data base administration **GNU**
 - data base analysis **GNU**
 - data base management **GNU**
 - data base operations **GNU**
 - field exit routine **GNU**
 - system installation **GNU**
- user table data **IRDU**
- user-maintained data **CALL**
- user-written application program **LLC**
 - description of **LLC**
 - functions of **LLC**
 - link-editing of **LLC**
 - INCLUDE statement **LLC**
 - module DLZDLI000 required **LLC**
 - requirements in relations to LLC/CC **LLC**
- user-written programming
 - application **DBA**
 - data base loading **DBA**
 - existing programs and packages **DBA**
 - other **DBA**
- USERCALL Type 1 (X'01') **DIAG**
- USERCALL Type 2 (X'02') **DIAG**
- USERID **GNU**
 - operand, BUILD statement **GNU, RDU**
 - using multiple logical relationships **GNU**
- USING PCB option **HLPI**
- using the interactive facilities **GNU**
- USREXIT operand of DLZTRACE **DIAG**
- USSA (unqualified SSA) **GNU**
 - format of **GNU**
- utilities **GNU**
 - choosing the right utilities **DBA**
 - data base backout **GNU, RRG**
 - data base change accumulation **GNU, RRG**
 - data base data set image copy **GNU**

data base data set recovery **GNU**
 Data Base Prefix Resolution **RRG**
 Data Base Prefix Update (DLZURGP0) **GNU, RRG**
 Data Base Preorganization (DLZURPRO) **GNU, RRG**
 Data Base Scan **RRG**
 Data Set Image Copy **RRG**
 Data Set Recovery **RRG**
 data-oriented **RRG**
 DL/I
 job stream generation in IUG **IRDU**
 DL/I recovery **DBA**
 DLZURGL0 (HD reorganization reload) **GNU**
 DLZURGS0 (data base scan) **GNU**
 DLZURGU0 (HD reorganization unload) **GNU**
 DLZURG10 (data base prefix resolution) **GNU**
 DLZURPR0 (data base preorganization) **GNU**
 DLZURRL0 (HISAM reorganization reload) **GNU**
 DLZURUL0 (HISAM reorganization unload) **GNU**
 EXTRACT DEFINES **GNU**
 file-oriented **RRG**
 HD Reorganization Reload (DLZURGL0) **GNU, RRG**
 HD Reorganization Unload (DLZURGU0) **GNU, RRG**
 HD Reorganization Unload/Reload **RRG**
 HISAM Reorganization Reload (DLZURRL0) **GNU, RRG**
 HISAM Reorganization Unload **RRG**
 HISAM Reorganization Unload (DLZURUL0) **GNU, RRG**
 HISAM Reorganization Unload/Reload **RRG**
 image copy **GNU**
 IUG job streams
 Backout **DIAG**
 Change Accumulation **DIAG**
 Extract Define **DIAG**
 HD Reorganization Reload **DIAG**
 HD Reorganization Unload **DIAG**
 HISAM Reorganization Reload **DIAG**
 HISAM Reorganization Unload **DIAG**
 Image Copy **DIAG**
 Initial Load **DIAG**
 Log Print **DIAG**
 Partial Reorganization, Part 1 **DIAG**
 Partial Reorganization, Part 2 **DIAG**
 Prefix Resolution **DIAG**
 Prefix Update **DIAG**
 Preorganization **DIAG**
 Recovery **DIAG**
 Scan **DIAG**
 Trace Print **DIAG**
 job streams
 data base load **IRDU**
 data base reorganization **IRDU**
 log print **GNU, RRG**
 overview **GNU**
 reorganization **GNU**
 storage requirements, DL/I data base **DBA**
 volume-oriented **RRG**
 VSAM VERIFY **GNU**
 when adding logical relationships, use of **DBA**
 utilities and serviceability aids **GIM**
 interactive utility generation **GIM**
 data base recovery utilities **GIM**
 data base reorganization/load processing utilities **GIM**
 automatic ISQL Extract Defines utility **GIM**
 serviceability aids **GIM**
 utility program problems **DIAG**
 utility programs **DBA, GNU, RDU**
 and data base loading **GNU**
 application control blocks creation and maintenance **DBA**
 data base backout **DBA, DIAG, IRDU, RDU**
 data base change accumulation **DBA, DIAG, IRDU, RDU**
 data base data set image copy **DBA, IRDU**

data base data set recovery **DBA, DIAG**
 data base image copy **DIAG, RDU**
 data base preorganization **DBA**
 data base prefix resolution **DBA, DIAG, IRDU, RDU**
 data base prefix update **DBA, DIAG, IRDU, RDU**
 data base preorganization **DBA, IRDU, RDU**
 data base recovery **DIAG, RDU**
 data base reorganization **DIAG**
 data base scan **DBA, DIAG, IRDU, RDU**
 data set image copy **DBA**
 data set recovery **IRDU**
 DL/I **GNU**
 HD reorganization reload **DIAG, IRDU, RDU**
 HD reorganization unload **DIAG, IRDU, RDU**
 HD unload **DBA**
 HISAM reorganization reload **DIAG, IRDU, RDU**
 HISAM reorganization unload **DIAG, IRDU, RDU**
 log print **DIAG, IRDU, RDU**
 logical relationship utilities **DIAG**
 partial data base reorganization **DIAG, IRDU, RDU**
 trace print **DBA, DIAG, IRDU**
 usage sequence **DBA**
 workfile generator **DIAG**

V

VALUE operand coding requirements **RDU**
 coding requirements **RDU**
 VIRFLD statement (PSBGEN) **GNU**
 variable length concatenated segment, length calculation **HLPI**
 variable length index source segments, considerations for **DBA**
 variable length segments **ADBD, CALL, DBA, CALL, GNU, HLPI**
 (see BYTES operand) **RDU**
 adding to **DBA**
 converting to **DBA**
 description **HLPI**
 in INSERT command **HLPI**
 in REPLACE command **HLPI**
 invalid length **CALL**
 length changes to **CALL**
 length field **CALL**
 processing **HLPI**
 residing in HD data bases **GNU**
 SEGLENGTH option **HLPI**
 size field **ADBD**
 specifying **GNU**
 starting position in **HLPI**
 uses **ADBD**
 VARIABLE option **HLPI**
 description **HLPI**
 if not specified **HLPI**
 INTO or FROM requirement **HLPI**
 segment I/O area **HLPI**
 with concatenated segment **HLPI**
 varying storage space **GNU**
 vendor name segment (STSIVND), description **GNU**
 verification **HLPI**
 combined checkpoint **HLPI**
 VERIFY command, access method services **GNU**
 VERIFY utility, VSAM **GNU**
 verifying open/close information **DIAG**
 view of the data base **GNU, HLPI**
 views of data **DBA**
 application **DBA**
 system **DBA**
 VIRFLD statement (PSBGEN) **GNU**
 BYTES operand **GNU**
 NAME operand **GNU**

RTNAME operand **GNU**
 START operand **GNU**
 TYPE operand **GNU**
 VALUE operand **GNU**
 VIRFLD statement format **RDU**
 virtual fields **CALL, DBA, GNU**
 defining **DBA**
 virtual insert rule, violation of **CALL**
 virtual logical child (VLC) **DBA, GNU**
 how to define **RDU**
 LCHILD, specification of **DBA**
 PARENT operand for **DBA**
 specification of logical child **DBA**
 virtual paired bidirectional logical relationship **GNU**
 virtual storage **GIM**
 virtual storage access method (VSAM) **GNU**
 virtual storage estimates, DL/I **DBA**
 virtual storage requirements worksheet, DL/I **DBA**
 VLC (virtual logical child) **GNU**
 volume copy **RRG**
 volume dump **RRG**
 volume restore **RRG**
 volume-oriented utilities **RRG**
 VSAM (virtual storage access method) **GNU, LLC**
 access method services **CALL**
 ALTER command **GNU**
 VERIFY command **GNU**
 DEFINE command **GNU**
 DELETE command **GNU**
 catalog **GNU**
 cluster concept **GNU**
 considerations
 in recovery/restart **GNU**
 control interval **GNU, LLC**
 BLOCK keyword **LLC**
 DATASET statement **LLC**
 data sets
 closing **GNU**
 definition **GNU**
 data spaces **GNU**
 defining clusters **GNU**
 DELETE run before DEFINE run **LLC**
 HIDAM data bases **LLC**
 master catalog **GNU**
 preparation of master catalog **LLC**
 requirements **GNU**
 sample job input stream **LLC**
 share options **GNU**
 3 and 4 **DIAG**
 VERIFY **GNU**
 run if data sets incorrectly closed **LLC**
 utility **GNU**
 VSAM asynchronous exit routine (DLZOVSEX) **DIAG**
 VSAM buffer allocation **CALL**
 VSAM buffers and options
 action modules **DBA**
 buffers **DBA**
 catalog **DBA**
 considerations in DL/I recovery/restart **DBA**
 control blocks **DBA**
 data sets, closing **DBA**
 options **DBA**
 real storage estimates **DBA**
 share option specification **DBA**
 VSAM data area **DIAG**
 VSAM DEFINE command **CALL**
 VSAM requirements, specifying **RDU**
 VSAM/SAM buffers **DBA**
 VSAMINTF (X'50') **DIAG**

VSE checkpoint **CALL, GNU, HLPI**
 for MPS restart facility **CALL, GNU, HLPI**
 in programming example **HLPI**
 restrictions **CALL, HLPI**
 VSE checkpoint/restart, restrictions **DBA**
 VSE CHKPT macro **RRG**
 VSE environment
 ACT job stream **IRDU**
 DBD job stream **IRDU**
 PSB job stream **IRDU**
 VSE Fast Copy Disk utility **RDU**
 VSE modules--access method dependent **DBA**
 VSE/Fast Copy Disk/Copy Disk Volume **RRG**

W

wait state debugging aids **DIAG**
 warm starts **RDU**
 MPS restart facility **RDU**
 WHAT NEXT entries
 description **IRDU**
 example **IRDU**
 WHERE option **GNU, HLPI**
 Boolean operators **GNU**
 operator in **HLPI**
 qualified segment selection **GNU**
 WHEREDBD **RDU**
 WHEREFLD **RDU**
 WHERESEG **RDU**
 workfile generator utility **DIAG**
 worksheets
 ACT **IRDU**
 DBD **IRDU**
 PSB **IRDU**
 Writing programs, techniques and suggestions **HLPI**
 WXTRNs **DIAG**

X

XDFLD statement
 format **RDU**
 of DBDGEN **CALL**
 XECB
 naming conventions **DIAG**
 requirements **DIAG**
 return codes **DIAG**
 table entries, locating **DIAG**
 usage cross-reference **DIAG**
 used in MPS **DIAG**
 waitlist dependence **DIAG**
 XECBTAB
 TYPE=CHECK **DIAG**
 TYPE=DEFINE **DIAG**
 TYPE=DELETE **DIAG**
 XOPTS options **HLPI**
 XPOST **DIAG**
 XWAIT **DIAG**

0

0P47A - log print utility **DIAG**

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/D¹
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/D²
For DL/I Version 1.4 ICR N/D²
For DL/I Version 1.5 N/D²
For DL/I Version 1.5 ICR N/D²

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/D³

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/D⁴
- Job Structure and Program Libraries N/D
- Status Code Checking N/D⁴
- Positive Segment Processing N/D⁴
- Online Application Program Design DBA
 - General Considerations DBA
 - Differences Between Batch, MPS, and Online DL/I DBA
 - Conventions N/D⁴
 - Reentrant Coding for DL/I N/D⁴
 - Use of DOS/VIS Communication Region N/D⁴
- Intent Scheduling vs Program Isolation DBA
 - DL/I Scheduling DBA
 - Program Isolation DBA
- Debugging N/D⁴
- Status Code Analysis N/D⁴
- Tracing N/D⁵
- 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/D⁴
 - 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/VIS Application Programming: High Level Programming Interface* or *DL/I DOS/VIS 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/D⁵
 - HISAM Logical Record Block Formats N/D⁵
 - HISAM Logical Record Block Formats N/D⁵
 - Hierarchical Direct - Logical Record and Block Formats N/D⁵
 - HDAM and HIDAM Storage Organizations DBA
 - HDAM DBA
 - HIDAM DBA
 - Format of Files Used for HDAM and HIDAM N/D⁵
 - 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/D⁶
- 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/D⁷
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 Preorganization 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/D⁸
Programming Systems Used for DL/I DOS/VS Version 1.4 ICR N/D⁸
Programming Systems Used for DL/I DOS/VS Version 1.5 N/D⁸
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/D⁹
 - Component Functions N/D⁹
- 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.

DL/I Application Control Table (ACT) DBA
CICS/VS - DL/I Tables Example RDU
DOS/VS Supervisor Generation RDU
MPS Operating Considerations DBA
MPS Sequence of Operations DBA
Storage Requirements DBA
MPS Performance Considerations DBA

Appendix A: Use and Creation of Pointers DBA

Physical Hierarchy Pointers
Physical Child First Pointer (PCF)
Physical Child Last Pointer (PCL)
Physical Twin Forward Pointer (PTF)
Physical Twin Backward Pointer (PTB)
Index Pointer
Logical Relationship Pointers
Physical Parent Pointer (PP)
Logical Parent Pointer (LP)
Logical Child First Pointer (LCF)
Logical Child Last Pointer (LCL)
Logical Twin Forward Pointer (LTF)
Logical Twin Backward Pointer (LTB)
Logical Child Counter (CTR)
Summary

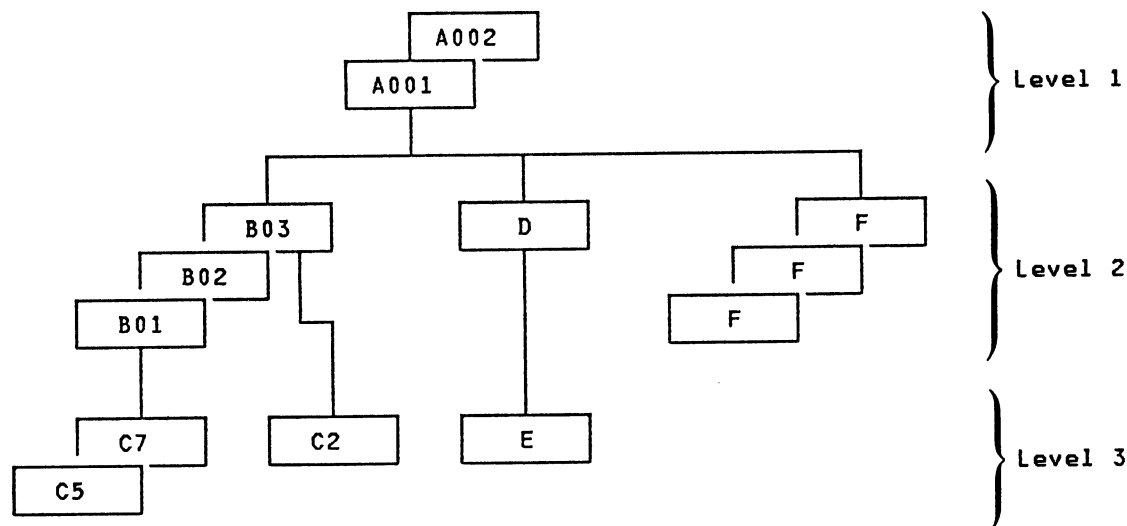
Appendix B: Manual Selection Guide LIB

Appendix C: DL/I in the CMS/DOS Environment DBA

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 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 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 Language/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

- ABDD** *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*

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? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Did you find the material: | | |
| Easy to read and understand? | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use? | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete? | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated? | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level? | <input type="checkbox"/> | <input type="checkbox"/> |
| • What is your occupation? | _____ | |
| • How do you use this publication: | | |
| As an introduction to the subject? | <input type="checkbox"/> | As an instructor in class? <input type="checkbox"/> |
| For advanced knowledge of the subject? | <input type="checkbox"/> | As a student in class? <input type="checkbox"/> |
| To learn about operating procedures? | <input type="checkbox"/> | As a reference manual? <input type="checkbox"/> |

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

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

Fold and Tape

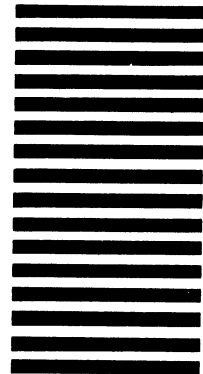
Please Do Not Staple

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 _____

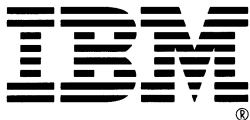
Company Name _____ Department _____

Street Address _____

City _____

State _____ Zip Code _____

IBM Branch Office serving you _____



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?	<input type="checkbox"/>	<input type="checkbox"/>		
• Did you find the material:				
Easy to read and understand?	<input type="checkbox"/>	<input type="checkbox"/>		
Organized for convenient use?	<input type="checkbox"/>	<input type="checkbox"/>		
Complete?	<input type="checkbox"/>	<input type="checkbox"/>		
Well illustrated?	<input type="checkbox"/>	<input type="checkbox"/>		
Written for your technical level?	<input type="checkbox"/>	<input type="checkbox"/>		
• What is your occupation?	_____			
• How do you use this publication:				
As an introduction to the subject?	<input type="checkbox"/>	As an instructor in class?	<input type="checkbox"/>	
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in class?	<input type="checkbox"/>	
To learn about operating procedures?	<input type="checkbox"/>	As a reference manual?	<input type="checkbox"/>	

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

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

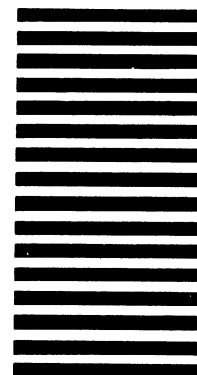
Fold and Tape

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 _____

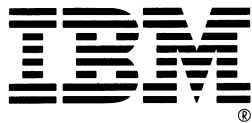
Company Name _____ Department _____

Street Address _____

City _____

State _____ Zip Code _____

IBM Branch Office serving you _____



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?	<input type="checkbox"/>	<input type="checkbox"/>		
• Did you find the material:				
Easy to read and understand?	<input type="checkbox"/>	<input type="checkbox"/>		
Organized for convenient use?	<input type="checkbox"/>	<input type="checkbox"/>		
Complete?	<input type="checkbox"/>	<input type="checkbox"/>		
Well illustrated?	<input type="checkbox"/>	<input type="checkbox"/>		
Written for your technical level?	<input type="checkbox"/>	<input type="checkbox"/>		
• What is your occupation?	_____			
• How do you use this publication:				
As an introduction to the subject?	<input type="checkbox"/>	As an instructor in class?	<input type="checkbox"/>	
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in class?	<input type="checkbox"/>	
To learn about operating procedures?	<input type="checkbox"/>	As a reference manual?	<input type="checkbox"/>	

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

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

Fold and Tape

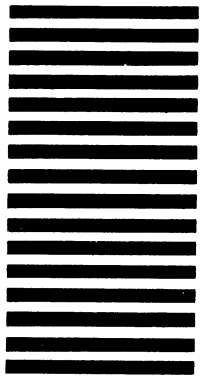
Please Do Not Staple

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 _____



GH24-5008-1

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



GH24-5008-01

