IBM Library Server Print Preview

DOCNUM = SC33-6211-04
DATETIME = 03/12/93 11:09:15
BLDVERS = 1.2
TITLE = IBM DL/I DOS/VS Release Guide
AUTHOR =
COPYR = © Copyright IBM Corp. 1984, 1993
PATH = /home/webapps/epubs/htdocs/book

COVER Book Cover

IBM Data Language/I
Disk Operating System/Virtual Storage

Release Guide

Version 1 Release 10

Document Number SC33-6211-04

Program Number
5746-XX1

File Number S370/S390-50

NOTICES Notices

```
 ___ Note! ________________________________________________________
|                                                                  |
| Before using this information and the product it supports, be sure |
| to read the general information under "Notices" in topic FRONT_1.  |
|                                                                  |
|__________________________________________________________________|
```

EDITION Edition Notice

Fifth Edition (March, 1993)

This edition applies to Version 1 Release 10 of IBM Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS), Program Number 5746-XX1, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

```
IBM Corporation      or to:        IBM Deutschland GmbH
Attn: Department ACV - BP           Department 3248
1001 WT Harris Boulevard            Schoenaicher Strasse 220
Charlotte, NC 28257,               D-7030 Boeblingen
U.S.A.                             Federal Republic of Germany
```

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

CONTENTS Table of Contents

Summarize

COVER **Book Cover**

NOTICES **Notices**

EDITION **Edition Notice**

CONTENTS **Table of Contents**

FIGURES **Figures**

FRONT_1 **Notices**
FRONT_1.1 **Programming Interface Information**
FRONT_1.2 **Trademarks and Service Marks**
FRONT_1.3 **Authorized Use of IBM Online Books**

ABSTRACT **Abstract**

PREFACE **About This Book**
  PREFACE.1 Who Should Use This Book
  PREFACE.2 How to Use This Book
  PREFACE.3 Where to Find More Information

CHANGES **Summary of Changes for DL/I**
CHANGES.1 DL/I 1.10 Enhancements
CHANGES.2 DL/I 1.9 Enhancements
CHANGES.3 DL/I 1.8 Enhancements

1.0 **Chapter 1. Operating Environment**
1.1 **Machine Requirements**
1.1.1 Virtual Storage
1.1.2 Disk Storage
1.1.3 Tape Drive
1.1.4 Terminal Support
1.2 **Running DL/I in a VM Environment**
1.3 **Programming Requirements**
1.3.1 DL/I 1.10
1.3.2 DL/I 1.9
1.3.3 DL/I 1.8
1.4 **Programming Environment**
1.4.1 Compiler and Assembler Support
1.4.2 Other Related Software Products

2.0 **Chapter 2. Migration and Compatibility Considerations**
2.1 **Migration from DL/I 1.7.0 to 1.7.1**
2.2 **Migration from DL/I 1.7.1 to 1.8.0**
2.3 **Migration from DL/I 1.8.0 to 1.9.0**
2.4 **Migration from DL/I 1.9.0 to 1.10.0**
2.4.1 DL/I Applications above the 16MB Line of Storage
2.4.2 VSE/VSAM HS-Buffers above the 16MB Line of Storage
2.4.3 Virtual Disk Exploitation
2.4.4 Change of Support for DL/I IMF and IUG
2.5 **Compatibility with IMS/VS**

3.0 **Chapter 3. Program Library Tape Contents**
3.1 Production Sublibrary
3.1.1 Source Books in the Production Sublibrary
3.2 Generation Sublibrary

4.0 **Chapter 4. Installation of DL/I**
4.1 **Installation on VSE/ESA, or on VSE Version 3 or 4**
4.2 **Installation on VSE/Advanced Functions 2.1 (or later)**

5.0 **Chapter 5. Customizing DL/I**
5.1 **SVA Loading for DL/I**
5.1.1 Loading all SVA Phases
5.1.2 Loading User-Specified SVA Phases
5.1.3 Loading SVA Phases via IPL
5.2 **CICS Adaptation**
5.3 **DL/I IMF and IUG Functions**
5.3.1 Deleting the VSE Version of IMF and IUG
5.3.2 Deleting the VM Version of IMF and IUG
5.3.3 Using IMF and IUG Functions in a VSE Environment
5.3.3.1 Accessing IMF and IUG Directly
5.3.3.2 Accessing IMF and IUG through an Existing ISPF Menu
5.3.4 Using IMF and IUG Functions in a VM Environment
5.3.4.1 Installing DL/I Dialogs on CMS
5.4 **DL/I Documentation Aid and the ISQL EXTRACT DEFINEs Utility**
5.4.1 DL/I Documentation Aid
5.4.2 ISQL EXTRACT DEFINEs

6.0 **Chapter 6. Post Installation Activities and Aids**
6.1 **How to Get the DL/I Sample Application Job Streams**
6.2 **MSHP Retrace**
6.3 **Displaying DL/I Library Directory Listings**
6.4 **DL/I Link Book**
6.5 **DL/I Merge Book**
6.6 **DL/I Delete Book**
6.7 **Debugging Aids**

7.0 **Chapter 7. Applying Service**
7.1 **Reassembling CICS Modules for Updated DL/I**
7.2 **Reassembling and Relinking DL/I for Updated CICS**
7.3 **Service for DL/I Application Interface Modules**
7.4 **Error Messages while Reassembling or Relinking DL/I Components**

8.0 **Chapter 8. New Features**
8.1 **New Features in DL/I 1.10**
8.2 **New Features in DL/I 1.9**
8.2.1 Support of VSE/ESA
8.3 **New Features in DL/I 1.8**
8.3.1 Date and Time on Reports and Statistics
8.3.2 Literal String in the HLPI WHERE Clause
8.3.3 Conditional Job Control Support
8.3.4 Status Code for Read-Only Programs
8.3.5 Partial Data Base Load
8.3.6 Increased Maximum Data Set Control Interval Size
8.3.7 Performance Improvements for Data Set Image Copy/Recovery
8.3.8 Device Independence
8.3.9 Automatic Verification for DL/I ESDS Data sets
8.3.10 Program Isolation Enhancement
8.4 **New Features in DL/I 1.7.1**
8.4.1 Improved Multiple Partition Support
8.4.2 Dynamically Scheduling MPS or Non-MPS Execution
8.4.3 The New GSTA Call
8.4.4 GSTA Call Format
8.4.5 GSTA Return Codes
8.4.6 Layout of the Statistics Buffer

9.0 **Chapter 9. DL/I 1.10 Support for VSE/ESA 1.3**
9.1 **DL/I Applications above the 16MB Line of Storage**
9.1.1 Invocation of Applications
9.1.2 COBOL II and COBOL Members
9.1.3 High Level Assembler and Assembler Members
9.1.4 Migrating Applications
9.1.4.1 Enabling COBOL Applications
9.1.4.2 Enabling Assembler Applications
9.1.5 Creating COBOL II Applications with 31-Bit Capability
9.1.5.1 Example: COBOL II - Online with HLPI or CALL Interface
9.1.5.2 Example: COBOL II - Batch and MPS Batch with HLPI
9.1.5.3 Example: COBOL II - Batch and MPS Batch with CALL Interface
9.1.6 Creating Assembler Applications with 31-Bit Capability
9.1.6.1 Example: High Level Assembler - Online
9.1.6.2 Example: High Level Assembler - Batch and MPS Batch
9.1.7 Placing DL/I Parameter Definitions Below the 16MB Line of Storage
9.1.7.1 CICS/VSE DL/I Online Application
9.1.7.2 DL/I Batch Application
9.2 **VSE/VSAM HS-Buffers above the 16MB Line of Storage**
9.2.1 Considerations
9.2.2 Compatibility of Existing Applications
9.2.3 Invocation
9.3 **Virtual Disk Exploitation**
9.4 **DL/I Run Statistics (DLZSTTL) Enhancements**

10.0 **Chapter 10. DL/I 1.9 Support for VSE/ESA 1.1**
10.1 **Dynamic Partitions**
10.2 **Database I/O Error Handling**
10.2.1 Repair of an I/O Error Data Base
10.3 **DL/I in a CICS/XRF Environment**
10.3.1 Requirements
10.3.2 Setup
10.3.3 System Integrity

11.0 **Chapter 11. New Messages**

A.0 **Appendix. Summary of Customer Interfacs**
A.1 DL/I Macros Intended for Customer Use

BIBLIOGRAPHY **Related IBM Manuals**
BIBLIOGRAPHY.1 IBM DL/I DOS/VS
BIBLIOGRAPHY.2 IBM VSE/ESA
BIBLIOGRAPHY.3 IBM VSE/SP Version 3
BIBLIOGRAPHY.4 IBM VSE/SP Version 4
BIBLIOGRAPHY.5 IBM VSE/ICCF
BIBLIOGRAPHY.6 IBM CICS
BIBLIOGRAPHY.7 IBM SQL/DS
BIBLIOGRAPHY.8 IBM ISPF
BIBLIOGRAPHY.9 IBM VS COBOL II
BIBLIOGRAPHY.10 IBM High Level Assembler

GLOSSARY **Glossary**

COMMENTS **Readers' Comments**

FIGURES Figures

1. Sample Job INSTALL for DL/I 1.8 4.2
2. Sample Job SVALOAD1 5.1.1
3. Sample Job SVALOAD2 5.1.2
4. Sample Job DELETVSE 5.3.1
5. Sample Job DELETVM 5.3.2
6. Sample Job ISPFCAT 5.3.3.2
7. Sample Job PUNCHDA 5.4.1
8. Naming Convention for DL/I DA and ISQL EXTRACT Modules 5.4.2
9. Sample Job PREP DLZDLBD 5.4.2
10. Sample Job CRTDLBDP 5.4.2
11. Sample Job DSPLYPRD 6.3
12. Sample Job DSPLYGEN 6.3
13. Sample Job PUNCH 7.1
14. Sample Job CICSGEN 7.1
15. Sample Job Stream ASSEM1 7.2
16. Sample Job Stream ASSEM2 7.2
17. Sample Job Stream ASSEM3 7.2
18. Sample Job Stream ASSEM4 7.2
19. Sample Job Stream HLASM 7.3
20. Sample Program DLZCTRL 8.4.2
21. How to Invoke the New GSTA Call in an Assembler Program 8.4.4
22. GSTA Call Return Codes 8.4.5
23. Layout of the Statistics Buffer DLZSTBF 8.4.6
24. Valid combinations of AMODE and RMODE 9.1.1
25. Example: COBOL II - Online with HLPI or Call Interface 9.1.5.1
26. Example: COBOL II - Batch and MPS Batch with HLPI 9.1.5.2
27. Example: COBOL II - Batch and MPS Batch with CALL 9.1.5.3
28. Example: High Level Assembler - Online 9.1.6.1
29. Example: High Level Assembler - Batch and MPS Batch 9.1.6.2
30. Re-allocation of DL/I Parameters in an Assembler Program 9.1.7.2
31. Example: Output of the DL/I Run Statistics Program (DLZSTTL) 9.4
32. Summary of Customer Interfaces A.1
33. Manuals in the DL/I DOS/VS Library BIBLIOGRAPHY.1

FRONT_1 Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressively designated by IBM, is the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577, U.S.A.

Subtopics:

- FRONT_1.1 Programming Interface Information
- FRONT_1.2 Trademarks and Service Marks
- FRONT_1.3 Authorized Use of IBM Online Books

FRONT_1.1 Programming Interface Information

This publication is intended to help the customer to install DL/I DOS/VS 1.10 (or 1.9 or 1.8). It contains installation, migration, customizing, and other post-installation information.

This publication also documents General-use Programming Interface and Associated Guidance Information, Product-sensitive Programming Interface and Associated Guidance Information, and Diagnosis, Modification or Tuning Information provided by DL/I DOS/VS.

General-use programming interfaces allow the customer to write programs that request or receive the services of DL/I DOS/VS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either as an introductory statement to a chapter or section or by the following marking:

------------------ General-use programming interface -------------------

General-use Programming Interface and Associated Guidance Information...

|--------------- End of General-use programming interface ---------------|

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of DL/I DOS/VS. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either as an introductory statement to a chapter or section or by the following marking:

```
 --------------- Product-sensitive programming interface ----------------

Product-sensitive Programming Interface and Associate Information...

|------------ End of Product-sensitive programming interface ------------|
```

Diagnosis, Modification or Tuning Information is provided to help the customer to do customization, diagnosis, modification, monitoring, repairing, tailoring, or tuning of DL/I DOS/VS.

Warning: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Diagnosis, Modification or Tuning Information is identified where it occurs, either as an introductory statement to a chapter or section or by the following marking:

```
 ------------ Diagnosis, Modification or Tuning Information -------------

Diagnosis, Modification or Tuning Information...

|--------- End of Diagnosis, Modification or Tuning Information ---------|
```

FRONT_1.2 Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

CICS	SQL/DS	VM/ESA
CICS/VSE	System/370	VM/XA
		VSE/ESA

FRONT_1.3 Authorized Use of IBM Online Books

For online versions of this book, we authorize you to:

° Copy, modify, and print the documentation contained on the media, for

ABSTRACT Abstract

This book documents the changes made for IBM DL/I 1.10:

° DL/I Applications above the 16MB line of storage
° VSE/VSAM HS-Buffers above the 16MB line of storage
° Virtual Disk Exploitation.

Furthermore, it includes a list of DL/I publications that are applicable with DL/I 1.10. Refer to "Related IBM Manuals" in topic BIBLIOGRAPHY.

This book is intended for persons responsible for setting up a DL/I system in an IBM VSE/ESA*****, VSE/SP, or VSE/Advanced Functions environment.

This book also includes information on installing, migrating, and customizing DL/I. It includes:

° Summaries of changes relating to previous releases of DL/I.

° Considerations for migrating from previous releases to DL/I 1.10.

° Messages that were changes or added in DL/I 1.9 and 1.10.

PREFACE About This Book

This book describes the installation of IBM DL/I DOS/VS:

- 1.10 on VSE/ESA*****,
- 1.9 on VSE/ESA, and
- 1.8 on VSE/SP or VSE/Advanced Functions.

It also includes information on migration and compatibility considerations, customizing, and post-installation tasks.

In this book, the term:

DL/I is used to refer to DL/I DOS/VS 1.8, 1.9, and 1.10.
*CICS****** is used to refer to CICS/DOS/VS 1.7, CICS/VSE***** 2.1, and CICS/VSE 2.2.

The full names are used only where necessary to distinguish.

Subtopics:

- PREFACE.1 Who Should Use This Book
- PREFACE.2 How to Use This Book
- PREFACE.3 Where to Find More Information

PREFACE.1 Who Should Use This Book

This book is for persons who are responsible for setting up a DL/I system in a VSE/ESA, VSE/SP, or VSE/Advanced Functions environment.

This book is written with the assumption that you have experience with the VSE environment in which you want to install DL/I, and that you are familiar with the concepts and terminology of DL/I.

PREFACE.2 How to Use This Book

- If you plan to *migrate* from a lower to a higher release level of DL/I, you should read Chapter 2, "Migration and Compatibility Considerations" in topic 2.0 before installing DL/I.

- If you plan to *install* DL/I on VSE/ESA, VSE/SP, or VSE/Advanced Functions, refer to Chapter 4, "Installation of DL/I" in topic 4.0.

For an *overview* on the new features of various releases of DL/I, refer to Chapter 8, "New Features" in topic 8.0. For further information on:

- DL/I 1.10, see Chapter 9, "DL/I 1.10 Support for VSE/ESA 1.3" in topic 9.0.

- DL/I 1.9, see Chapter 10, "DL/I 1.9 Support for VSE/ESA 1.1" in topic 10.0.

- DL/I 1.8, see "New Features in DL/I 1.8" in topic 8.3. Refer also to the manual *DL/I Resource Definition and Utilities* (which is completely revised for release 1.8).

- DL/I 1.7.1, see "New Features in DL/I 1.7.1" in topic 8.4.

PREFACE.3 Where to Find More Information

For a list of the DL/I publications and other related manuals, refer to "Related IBM Manuals" in topic BIBLIOGRAPHY.

CHANGES Summary of Changes for DL/I

The following summarizes the changes made to DL/I.

Subtopics:

- CHANGES.1 DL/I 1.10 Enhancements
- CHANGES.2 DL/I 1.9 Enhancements
- CHANGES.3 DL/I 1.8 Enhancements

CHANGES.1 DL/I 1.10 Enhancements

- Support for VSE/ESA*****:
  - DL/I applications above the 16MB line of storage
  - VSE/VSAM HS-buffers above the 16MB line of storage
  - Virtual disk exploitation

- Other changes:
  - Improved DL/I run statistics (APAR PN21734)
  - Improved handling of database I/O errors (APAR PN09116)

CHANGES.2 DL/I 1.9 Enhancements

- Support of VSE/ESA
  - CICS/XRF (Extended Recovery Facility)

   - Dynamic Partitions

- ° Device Independence

- ° Automatic Verification for DL/I ESDS Data Sets

- ° Program Isolation Enhancement

CHANGES.3 DL/I 1.8 Enhancements

- ° Device Independence (APAR PL48345)

- ° Automatic Verification for DL/I ESDS Data Sets (APAR PL20988)

- ° Program Isolation Enhancement (APAR PL55587)

1.0 Chapter 1. Operating Environment

This chapter describes the machine and programming requirements as well as the programming environment for the operation of DL/I DOS/VS.

Subtopics:

- 1.1 Machine Requirements
- 1.2 Running DL/I in a VM Environment
- 1.3 Programming Requirements
- 1.4 Programming Environment

1.1 Machine Requirements

Subtopics:

- 1.1.1 Virtual Storage
- 1.1.2 Disk Storage
- 1.1.3 Tape Drive
- 1.1.4 Terminal Support

1.1.1 Virtual Storage

The minimum main storage requirements for DL/I are the same as those needed for conventional operation of VSE/ESA*****, VSE/SP, or VSE/Advanced Functions. Because of virtual storage support, the minimum configuration is dependent on application characteristics and performance requirements. Processor performance may be traded off against main storage size.

For additional details regarding storage needs, refer to the manual *DL/I Data Base Administration*, in:

"Part: Installation Planning"
"Appendix: DL/I Virtual Storage Estimate"
"Appendix: DL/I Real Storage Estimate"

1.1.2 Disk Storage

The approximate library space (in library blocks) needed for DL/I installation is given below. Either check that an existing sublibrary has enough free library blocks, or allocate a new library of at least the indicated size.
It is advisable to allocate 5% to 10% additional free space for adding new code and applying service.

	Production Sublibrary	Generation Sublibrary
DL/I on VSE	12000	13000

To get the actual amount of library space needed for DL/I installation, scan the distribution tape using the VSE provided installation dialog.

1.1.3 Tape Drive

You need at least one tape drive for the installation of DL/I.

Using the IBM 3480 or 3490 as Logging Device: When you use the IBM 3480 or IBM 3490 magnetic tape subsystem as a DL/I log tape, you have to use *Tape Write Immediate* (non-buffered) mode. This is for reasons of data integrity when the IBM 3480 or IBM 3490 is used as a DL/I log tape. It ensures that data in the buffer are not lost if a power interruption or system failure occurs.

Also, CICS/VSE***** 2.2 supports the IBM 3480 and IBM 3490 magnetic tape subsystems. However, these units are *not* recommended for journaling.

1.1.4 Terminal Support

Any terminal device supported by CICS*, or equivalent product, may be used for online DL/I. Use of the DL/I IMF and IUG functions requires a 3270-type terminal.

1.2 Running DL/I in a VM Environment

If you plan to run VSE/ESA, VSE/SP, or VSE/Advanced Functions under VM, you have to consider VM storage requirements.

For using DL/I with CICS in a VM virtual machine (under control of VSE), the following considerations apply:

- CICS when operating in a virtual machine has the same hardware and software requirements as CICS operating in a real machine. Other software components (for example, access methods, compilers, and the release of VSE under which CICS runs) must be valid for the release of CICS you are using.

- The minimum hardware requirements of CICS should be considered as additional to the minimum requirements for VM itself and any other virtual machines within the VM environment.

- CPU utilization and possibly terminal response times will be greater when CICS is running under VM than when it is running in a real machine environment. The effect on performance will be most noticeable when VM is introduced into an installation where CPU and main storage resources are already substantially committed to existing CICS and other work.

1.3 Programming Requirements

The following licensed IBM programs are required for the operation of DL/I:

Subtopics:

- 1.3.1 DL/I 1.10
- 1.3.2 DL/I 1.9
- 1.3.3 DL/I 1.8

1.3.1 DL/I 1.10

```
 _________________________________________________________________________
|                                                                         |
| °   VSE/ESA 1.3 (5750-ACD) (or later)                                   |
|                                                                         |
| °   DOS/VS Sort/Merge II 2.5 (5746-SM2) or equivalent                   |
|                                                                         |
|_________________________________________________________________________|
```

1.3.2 DL/I 1.9

```
 _________________________________________________________________________
|                                                                         |
| °   VSE/ESA 1.1 (5750-ACD) (or later)                                   |
|                                                                         |
| °   DOS/VS Sort/Merge II 2.5 (5746-SM2) or equivalent                   |
|                                                                         |
|_________________________________________________________________________|
```

1.3.3 DL/I 1.8

```
 ___ Either: _____________________________________________________________
|                                                                         |
| °   VSE/SP 3.1 (5666-345) (or later)                                    |
|                                                                         |
| °   DOS/VS Sort/Merge II 2.5 (5746-SM2) or equivalent                   |
|                                                                         |
|_________________________________________________________________________|

 ___ Or: _________________________________________________________________
|                                                                         |
| °   VSE/Advanced Functions 2.1 (5666-301) (or later).                   |
|                                                                         |
|     To install DL/I 1.8 on VSE/Advanced Functions 2.1, the              |
|     installation of PTF UD36951 is prerequisite.  This PTF is           |
|     integrated in the VSE/Advanced Functions 2.1.6 refresh release.     |
|                                                                         |
| °   VSE/VSAM 1.3 (5746-AM2) (or later)                                  |
|                                                                         |
| °   DOS/VS Sort/Merge II 2.5 (5746-SM2) or equivalent                   |
|                                                                         |
|_________________________________________________________________________|
```

1.4 Programming Environment

DL/I is designed to work with the following licensed IBM programs:

° **CICS**:

  - **CICS/VSE* 2.2** (with DL/I 1.10)

- **CICS/VSE* 2.1** (with DL/I 1.9)
- **CICS/DOS/VS 1.7** (with DL/I 1.8)

CICS is required if you plan to use any of the following:

- Online applications
- High Level Programming Interface (HLPI) functions in a batch or online environment
- Multiple Partition Support (MPS)

° **VSE/VSAM Space Management for SAM Feature 1.1** or later.

° **Interactive System Productivity Facility (ISPF) 1.1** or later.

Note: VSE/ESA no longer supports ISPF, a prerequisite for DL/I IMF and IUG. For more information refer to "DL/I IMF and IUG Functions" in topic 5.3.

° **VSE/ICCF 2.1** or later.

DL/I is designed to run in batch mode or in Multiple Partition Support (MPS) batch mode in a VSE batch partition or in VSE/ICCF pseudo partitions. VSE/ICCF can also be the host environment for ISPF, under which DL/I IMF and IUG run on a VSE/SP or VSE/Advanced Functions system.

° **Structured Query Language/Data System (SQL/DS*) 1.3** or later.

° **VM/SP 3.1** or later.

VM (including VM/XA* and VM/ESA*) supports the DL/I IMF and IUG functions through ISPF.

Subtopics:

- 1.4.1 Compiler and Assembler Support
- 1.4.2 Other Related Software Products

1.4.1 Compiler and Assembler Support

DL/I is supported for use with the following compilers and assemblers:

° **VS COBOL II**:

- Library
- Compiler and Library
- Compiler, Library, and Debug

- **DOS/VS COBOL**:
  - Library
  - Compiler and Library
- **DOS PL/I**
  - Optimizing Compiler
  - Resident Library
  - Transient Library
- **DOS/VS RPG II 1.3**
- **High Level Assembler**
- **Assembler**

1.4.2 Other Related Software Products

- **Data Management System/CICS**
  Helps to inquire, insert, update, and delete database records. Access into the database may be via primary or secondary indexes.
- **DL/I DOS/VS Space Management Utilities (Installed User Program)**
  Help improve system performance and programmer productivity. They detect and report DL/I HD (hierarchical direct) pointer discrepancies, provide statistics and information for HD tuning, and assist with segment restructuring and reloading during database reorganization.
- **Cross System Product/Application Development** and
  Cross System Product/Application Execution
  Simplify the development and execution of DL/I application program that maintain or use DL/I databases.
- **Query DL/I**
  is an optional product which provides query facilities for operational DL/I databases. With its easy-to-use menus, it offers functions to get up-to-the-minute information.

2.0 Chapter 2. Migration and Compatibility Considerations

This chapter discusses considerations for DL/I users who want to migrate from a previous release up to DL/I 1.10.

Depending on your current DL/I release, select the appropriate entry point:

Your Current DL/I Release	Start with Entry Point
1.7.0	"Migration from DL/I 1.7.0 to 1.7.1" in topic 2.1
1.7.1	"Migration from DL/I 1.7.1 to 1.8.0" in topic 2.2
1.8.0	"Migration from DL/I 1.8.0 to 1.9.0" in topic 2.3
1.9.0	"Migration from DL/I 1.9.0 to 1.10.0" in topic 2.4

Furthermore, this chapter discusses the compatibility of DL/I with IMS/VS.

Subtopics:

- 2.1 Migration from DL/I 1.7.0 to 1.7.1
- 2.2 Migration from DL/I 1.7.1 to 1.8.0
- 2.3 Migration from DL/I 1.8.0 to 1.9.0
- 2.4 Migration from DL/I 1.9.0 to 1.10.0
- 2.5 Compatibility with IMS/VS

2.1 Migration from DL/I 1.7.0 to 1.7.1

DL/I 1.7.1 is upwardly compatible from DL/I 1.7.0.

Changes are required to DL/I user application programs only for the following cases:

° The DL/I call function GSCD is supported only for applications operating in the same non-shared address space as DL/I. Applications operating in address space other than the one in which DL/I and CICS* are operating must be converted to use the new GSTA call to retrieve the buffer statistics.

° ACBGEN must be rerun for any PSBs for which you wish to issue the new GSTA calls.

° "PATH" sensitivity must be specified for all segments for which you do path inserts from MPS jobs running in separate address spaces. Currently, "PATH" sensitivity is required only for path retrieves but not for path inserts. (This is also a requirement for users of remote PSB support.)

Except for the above mentioned, no other ACBGEN, DBDGEN, or PSBGEN run is required for existing programs.

Continue the migration with "Migration from DL/I 1.7.1 to 1.8.0" in topic 2.2.

2.2 Migration from DL/I 1.7.1 to 1.8.0

DL/I 1.8.0 is upwardly compatible from DL/I 1.7.1.

Changes are required to DL/I user application programs only for the following cases:

- All online users must reassemble the online nucleus with the DL/I DLZACT macro.
- All programs which contain DLZTRACE macro calls with OUTPUT=SYSLST must be reassembled and re-linked.
- All user-written modules or programs (for example, randomizing routines) which reference the Partition Specification Table (PST) must be reassembled and re-linked.

Continue the migration with "Migration from DL/I 1.8.0 to 1.9.0" in topic 2.3.

2.3 Migration from DL/I 1.8.0 to 1.9.0

DL/I 1.9 is upwardly compatible from DL/I 1.8.0.

Changes are required to DL/I user application programs only for the following cases:

- All online users must reassemble the online nucleus with the DL/I DLZACT macro.
- All online users must reassemble and re-link their online trace programs using the DLZTRACE macro.

Continue the migration with "Migration from DL/I 1.9.0 to 1.10.0" in topic 2.4.

For DL/I 1.9, the explanation given under "Change of Support for DL/I IMF and IUG" in topic 2.4.4 also applies.

2.4 Migration from DL/I 1.9.0 to 1.10.0

DL/I 1.10 is upwardly compatible from DL/I 1.9.0.

If you are an online user migrating from DL/I 1.9 to DL/I 1.10, you have to reassemble the online nucleus with the DL/I DLZACT macro.

If you want to exploit the support provided with DL/I 1.10, you have to make changes to your existing applications and procedures as described below.

Subtopics:

- 2.4.1 DL/I Applications above the 16MB Line of Storage
- 2.4.2 VSE/VSAM HS-Buffers above the 16MB Line of Storage
- 2.4.3 Virtual Disk Exploitation
- 2.4.4 Change of Support for DL/I IMF and IUG

2.4.1 DL/I Applications above the 16MB Line of Storage

If you want to *enable* an existing DL/I application (COBOL or assembler) for execution in address space above the 16MB line of storage, proceed as explained under "Migrating Applications" in topic 9.1.4.

2.4.2 VSE/VSAM HS-Buffers above the 16MB Line of Storage

If you want to allocate VSE/VSAM input/output buffers above the 16MB line of storage for:

- KSDS index files, or
- HISAM KSDS and ESDS data files, or
- SHISAM KSDS data files,

proceed as explained under "VSE/VSAM HS-Buffers above the 16MB Line of Storage" in topic 9.2.

2.4.3 Virtual Disk Exploitation

If you want the work files of *DL/I utilities* to reside on virtual disk instead of on a real device, the job control statements (ASSGN, DLBL and EXTENT) must address the virtual disk instead of the real device.

For more information, refer to "Virtual Disk Exploitation" in topic 9.3.

2.4.4 Change of Support for DL/I IMF and IUG

The Interactive System Productivity Facility (ISPF), a prerequisite for using the DL/I IMF and IUG functions, is not supported by VSE/ESA*****.

Users operating VSE/ESA under the control of VM/SP, VM/XA*****, or VM/ESA***** can continue to use DL/I IMF and IUG through ISPF running on CMS.

2.5 Compatibility with IMS/VS

DL/I is compatible with IMS/VS for the batch user and to IMS/VS with a CICS***** subsystem for the online user, with the following exceptions:

° The following facilities in DL/I are not supported by IMS/VS:

- HD or HISAM UNLOAD in IMS/VS and RELOAD in DL/I DOS/VS
- RPG-II support
- FBA devices
- CKD device independence
- CALLDLI MF=E Assembler Language DL/I programs
- IMF and IUG
- Disk logging
- Application control table (ACT)
- ACCESS macro
- Extended remote PSB
- Selective UNLOAD
- CICS Monitoring Facility (CMF) DL/I PA replacement
- Run and buffer statistics
- Checkpoint option for HD UNLOAD
- Rewind option in utilities
- Documentation Aid
- MPS Restart
- Variable-length index source segments
- PL/I options
- GSTA call

° The following DL/I facilities are not fully compatible or are supported by IMS/VS in a different manner and require some user modifications:

- High level programmer interface (IMS/VS)
- Log files
- Image copy files
- Index format
- Field level sensitivity
- HD UNLOAD in DL/I DOS/VS and HD RELOAD in IMS/VS
- HISAM UNLOAD in DL/I DOS/VS and HISAM RELOAD in IMS/VS
- UNLOAD/RELOAD disk files
- CKPT CALL parameters
- MODEL=3330-II
- Default PSB on scheduling call
- DL/I status code NI
- Control Interval sizes
- Partial Data Base Load

For further information on IMS/VS compatibility refer to the manual *DL/I Data Base Administration*, in "Appendix: Incompatibilities between DL/I and

IMS."

3.0 Chapter 3. Program Library Tape Contents

DL/I DOS/VS is distributed as a RYO distribution tape. This tape is part of the tapes that contain the optional products offered with VSE/ESA***** and VSE/SP.

The DL/I RYO distribution tape is assembled with:

- CICS/VSE***** 2.2. for DL/I 1.10 (CLC=DB5)
- CICS/VSE***** 2.1. for DL/I 1.9 (CLC=1EF)
- CICS/VSE***** 1.7. for DL/I 1.8 (CLC=C59)

The tape can be restored to disk through the MSHP INSTALL function.

The tape contains a complete set of (or replacement for) the DL/I libraries, including all IMF and IUG functions, panels, skeletons, and message files.

Note: The VSE version of DL/I IMF and IUG is not included with DL/I 1.10.

DL/I is partially delivered as object code only.

DL/I is shipped with a production part and a generation part as follows.

Subtopics:

- 3.1 Production Sublibrary
- 3.2 Generation Sublibrary

3.1 Production Sublibrary

The DL/I production sublibrary contains:

- All DL/I phases (type PHASE)

- All DL/I modules (type OBJ)

- All primary source code of DL/I, consisting of:

  - Source books of type A
  - All source books of type E
  - All IMF and IUG source books of type M, N, and S (not for DL/I 1.10)
  - Sample programs of type A, C, P, and R (see "Source Books in the

Production Sublibrary" in topic 3.1.1, below)

Subtopics:

- 3.1.1 Source Books in the Production Sublibrary

3.1.1 Source Books in the Production Sublibrary

The DL/I production sublibrary contains the following source books needed for certain tasks:

Structure for User Interface Blocks

```
DLIUIB.C  User Interface Block for COBOL
DLIUIB.P  User Interface Block for PL/I
DLIUIB.R  User Interface Block for RPG
```

DL/I Assembler Sample Programs

```
DLZMAP.A  Mapping module for DLZSAM60
DLZSAMAC.A Batch ACCESS sample application job stream
DLZSAMJS.A Online sample application job stream
DLZSAM40.A Load program
DLZSAM50.A Batch print program
DLZSAM60.A Online sample application program
```

DL/I High Level Assembler Sample Programs (as of DL/I 1.10)

```
DLZHMAP.A Mapping module for DLZHLA60
DLZHLA40.A Load program
DLZHLA50.A Batch print program
DLZHLA60.A Online sample application program
```

DL/I COBOL II Sample Programs (as of DL/I 1.10)

```
DLZCB2MP.A Mapping module for DLZCB230, DLZCB260
DLZCB210.A HLPI load program
DLZCB220.A HLPI batch print program
DLZCB230.A Online HLPI sample application program
DLZCB240.A Load program
DLZCB250.A Batch print program
DLZCB260.A Online sample application program
```

DL/I COBOL Sample Programs

```
DLZCBMAP.A Mapping module for DLZCBL30, DLZCBL60
DLZCBL10.A HLPI load program
DLZCBL20.A HLPI batch print program
DLZCBL30.A Online HLPI sample application program
```

```
DLZCBL40.A Load program
DLZCBL50.A Batch print program
DLZCBL60.A Online sample application program
```

DL/I PL/I Sample Programs

```
DLZPLMAP.A Mapping module for DLZPLI30, DLZPLI60
DLZPLI10.A HLPI load program
DLZPLI20.A HLPI batch print program
DLZPLI30.A Online HLPI sample application program
DLZPLI40.A Load program
DLZPLI50.A Batch print program
DLZPLI60.A Online sample application program
```

DL/I RPG Sample Programs

```
DLZRGMAP.A Mapping module for DLZRPG60
DLZRPG40.A Load program
DLZRPG50.A Batch print program
DLZRPG60.A Online sample application program
```

DL/I Documentation Aid, ISQL EXTRACT DEFINEs (1)

```
DLZDATAB.A Acquires DB space and creates the SQL/DS* DL/I DA tables
DLZDANDX.A Creates the SQL/DS DL/I DA table indexes
DLZDARTN.A DATALOADs the ISQL DL/I DA routines
DLZDLBD.A Creates the DBD ACCESS module
DLZDLBP.A Creates the PSB ACCESS module
DLZEXDF.A Job stream for installing ISQL EXTRACT DEFINEs
DLZEXWCB.A EXTRACT DEFINEs work control block
DLZSQLID.A USERID control block
```

Special DL/I Books

```
DLZACTDS.A DSECT for application control table (ACT)
DLZDBGLB.A COPY book for DBD global values
DLZDLETE.A DL/I delete book
DLZLNKBK.A DL/I link book
DLZMERGE.A DL/I merge book
```

---

(1) Support for DL/I Extract has been removed by SQL/DS* Version 3.

3.2 Generation Sublibrary

The DL/I generation sublibrary contains:

- The VM Version of IMF and IUG (DLZCMSTL.Z and DLZCMSML.Z).
- The optional source books of DL/I (type A for DL/I module and macro source code).

4.0 Chapter 4. Installation of DL/I

This chapter describes the installation of DL/I on:

- VSE/ESA***** or VSE/SP
  Refer to "Installation on VSE/ESA, or on VSE Version 3 or 4" in topic 4.1.
- VSE/Advanced Functions
  Refer to "Installation on VSE/Advanced Functions 2.1 (or later)" in topic 4.2.

Subtopics:

- 4.1 Installation on VSE/ESA, or on VSE Version 3 or 4
- 4.2 Installation on VSE/Advanced Functions 2.1 (or later)

4.1 Installation on VSE/ESA, or on VSE Version 3 or 4

For the installation of the DL/I distribution tape use the VSE installation dialogs described in *VSE/SP Installation* or in *VSE/ESA Installation and Service*.

4.2 Installation on VSE/Advanced Functions 2.1 (or later)

After having installed VSE/Advanced Functions, follow the instructions in the applicable VSE/SP-version manual *VSE/SP Installation*.

The following lists DL/I specific installation considerations that are not covered in that manual:

1. Ensure that label IJSYS02 is available for a compiler work file. Also ensure sufficient file size, for example 1000 FBA blocks or 20 tracks (IBM 3380).
2. Ensure that two labels for the target libraries (here: DLIPRD and DLIGEN) are defined for the installation of DL/I. For the recommended minimum size, see "Disk Storage" in topic 1.1.2.

Note: You can install the production sublibrary and the generation sublibrary in two separate sublibraries (as shown in this book), or you can install both sublibraries together in one sublibrary. You can also choose between overwriting old sublibraries and creating new sublibraries for the DL/I base library.

The library names DLIPRD (for the DL/I production sublibrary) and DLIGEN (for the DL/I generation sublibrary) are chosen arbitrarily and are used in the sample job stream Figure 1.

3. Use the following job to install the DL/I base library. You can either enter the job through the reader or type it in at the console. In Figure 1:

   ° Do not misspell the ID "DL/I-BASE..1.8," as it will be used to position the tape.
   ° SYS006 is assigned to the distribution tape.
   ° *cuu* is the tape unit address of the distribution tape.

```
// JOB INSTALL
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid,1,0,xxxx,yyyy
// DLBL DLIGEN,'DL/I-generation-library'
// EXTENT ,volid,1,0,xxxx,yyyy
// ASSGN SYS006,cuu
// MTC REW,SYS006
// EXEC MSHP
INSTALL PRODUCT FROMTAPE ID='DL/I-BASE..1.8' -
        PRODUCTION INTO=DLIPRD.sublib -
        GENERATION INTO=DLIGEN.sublib;
/*
/&
```

Figure 1. Sample Job INSTALL for DL/I 1.8

5.0 Chapter 5. Customizing DL/I

Subtopics:

- 5.1 SVA Loading for DL/I
- 5.2 CICS Adaptation
- 5.3 DL/I IMF and IUG Functions
- 5.4 DL/I Documentation Aid and the ISQL EXTRACT DEFINEs Utility

5.1 SVA Loading for DL/I

The following DL/I phases are eligible for residence in the SVA:

$SVADLI	SVA Load List
DLZCPY10	Field Level Sensitivity Copy
DLZDBH00	Buffer Handler
DLZDLA00	Call Analyzer
DLZDLD00	Delete/Replace
DLZDLR00	Retrieve
DLZDDLE0	Load/Insert
DLZDHDS0	Space Management
DLZDXMT0	Index Maintenance
DLZSTRB0	Batch Field Storage Manager

Together, these DL/I phases require a physical size of at least 116K in the SVA. You have to reserve the space during system start-up by using the IPL command:

```
     SVA ....,PSIZE=nnnK,...
```

You can load all or a selection of these phases into the SVA. Proceed as shown under:

- "Loading all SVA Phases" in topic 5.1.1
- "Loading User-Specified SVA Phases" in topic 5.1.2
- "Loading SVA Phases via IPL" in topic 5.1.3

Subtopics:

- 5.1.1 Loading all SVA Phases
- 5.1.2 Loading User-Specified SVA Phases
- 5.1.3 Loading SVA Phases via IPL

5.1.1 Loading all SVA Phases

To load *all* SVA-eligible phases, use the SVA load list $SVADLI and run a job similar to that shown in Figure 2.

```
      // JOB SVALOAD1
      // DLBL DLIPRD,'DL/I-production-library'
      // EXTENT ,volid
      // LIBDEF PHASE,SEARCH=DLIPRD.sublib
      SET SDL
      LIST=$SVADLI
      /*
      /&
```

Figure 2. Sample Job SVALOAD1.
Use this job to load **all** SVA-eligible DL/I phases into the SVA.

5.1.2 Loading User-Specified SVA Phases

To load *selected* SVA-eligible phases, run a job similar to that shown in Figure 3.

```
// JOB SVALOAD2
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// LIBDEF PHASE,SEARCH=DLIPRD.sublib
SET SDL
DLZCPY10,SVA
DLZDBH00,SVA
DLZDLA00,SVA
DLZDLD00,SVA
DLZDLR00,SVA
DLZDDLE0,SVA
DLZDHDS0,SVA
DLZDXMT0,SVA
DLZSTRB0,SVA
/*
/&
```

Figure 3. Sample Job SVALOAD2.
Use this job to load **selected** SVA-eligible DL/I phases into the SVA.

5.1.3 Loading SVA Phases via IPL

To load SVA-eligible DL/I phases *automatically* each time IPL is performed, you may incorporate the appropriate job control statements into your BG ASI procedure $0JCL.

Depending on whether you want to include all or user-selected phases, use the job control statements shown in:

- Figure 2 in topic 5.1.1
- Figure 3 in topic 5.1.2

In the load procedure, omit the // JOB and /& statements.

5.2 CICS Adaptation

If you plan to:

- run DL/I online,
- employ MPS, or
- use HLPI functions,

ensure that the interface between DL/I and CICS* is correct by performing the following steps:

1. Define DL/I databases and application programs during the preparation of CICS tables.

a. In the CICS file control table (FCT), include an entry for each database descriptor (DBD) corresponding to a physical database. The name in the DATASET parameter in the FCT and the NAME parameter in the DBD must be identical.

b. Define the applications that access the DL/I databases in your PCT and PPT (if you are using tables), or your CSD (if you are using RDO).

2. Specify DL/I and CICS system table macros for DL/I support as follows:

a. Define the DL/I application control table (ACT). This table is required to associate online application programs with one or more DL/I databases.

b. Optionally define a storage layout control (SLC) table for use in an online environment, to specify the sequence in which DL/I modules are to be loaded from the DL/I production sublibrary during DL/I initialization.

c. If program isolation is active or emergency restart or dynamic transaction backout is to be used with DL/I tasks, assign the DL/I log to the CICS system log. The DL/I log is assigned by use of the VSE UPSI byte information. Refer to the following books for information on how to use this UPSI byte with CICS:

- *CICS/VSE 2.2 System Definition and Operations Guide*
- *&bcsdopx.*
- *CICS/DOS/VS 1.7 Installation and Operations Guide*
- *DL/I Guide for New Users*, under "Chapter: Online and MPS Considerations, UPSI Byte Settings (Online)"

d. If you are using the execution diagnostic facility with the application programs containing EXEC DLI commands, the DL/I language definition table (DLZHLPI) must be known to CICS. DLZHLPI is a module provided with DL/I DOS/VS.

  - If you are using RDO, DLZHLPI is defined for you in the CSD by the DFHCSDUP INITIALIZE command. It is also included in the group list DFHLIST.

  - If you are not using RDO, you must define DLZHLPI in the processing program table (PPT) by using the DFHPPT TYPE=ENTRY macro instruction.

e. If you want to capture DL/I run and buffer statistics, you must include the DL/I module DLZSTTL in your resource definitions. The run and buffer statistics function captures online DL/I system statistics and writes them to the extra-partition CICS statistics destination CSSL. This data is automatically printed during CICS shutdown, or printing can be invoked by the CSDE transaction. You can define DLZSTTL to your CICS system in one of two ways:

  1) If you are using RDO, use the CEDA transaction to define DLZSTTL in a suitable group in your CSD.

  2) If you are not using RDO, specify DLZSTTL in your PPT and PCT as follows:

```
DFHPCT TYPE=ENTRY,PROGRAM=DLZSTTL,TRANSID=CSDE
DFHPPT TYPE=ENTRY,PROGRAM=DLZSTTL
```

f. The CICS Monitoring Facilities (CMF) let you, as a CICS DL/I user, collect performance data during online processing. For detailed information, refer to the manual *DL/I Data Base Administration*, in "Chapter: Monitoring the Data Base," under "CICS/DOS/VS Monitoring Facilities (CMF) Hooks."

A CICS-DL/I installation tester (DFHTDLI) application program is supplied with the pre-generated CICS system. For the description of the program, refer to the manual:

- *CICS/VSE 2.2 System Definition and Operations Guide*.

- *&bcsdopx.*.

- *CICS/DOS/VS 1.7 Installation and Operations Guide*.

For a full description of how to define and generate the DL/I tables, refer to the manual *DL/I Resource Definition and Utilities*.

For a description of how to use DL/I and CICS XRF, refer to Chapter 10, "DL/I 1.9 Support for VSE/ESA 1.1" in topic 10.0.

5.3 DL/I IMF and IUG Functions

```
 ___ Note __________________________________________________________________
|                                                                          |
| ISPF, a prerequisite for using DL/I IMF and IUG, is not supported by     |
| VSE/ESA*.                                                                |
|                                                                          |
| However, DL/I IMF and IUG functions can be used when ISPF is running     |
| under:                                                                   |
|                                                                          |
| °   VSE/SP or VSE/Advanced Functions, or                                 |
| °   VM/CMS.                                                              |
|                                                                          |
| The following information applies under these conditions.               |
|                                                                          |
|__________________________________________________________________________|
```

DL/I provides two interactive functions: the Interactive Macro Facility (IMF), and the Interactive Utility Generation (IUG) facility. These functions offer easy-to-use interactive procedures that let you perform resource definition and utility functions at a terminal. For a full description of these DL/I functions, refer to the manual *DL/I Interactive Resource Definition and Utilities*.

The Interactive System Productivity Facility (ISPF) handles the menus and dialogs for DL/I IMF and IUG.

Note: If you intend to use IMF tables which were originally created under IPF (Interactive Productivity Facility) with a DL/I release prior to DL/I 1.7, these tables must be converted to ISPF format. To perform the conversion, refer to the manual *DL/I Interactive Resource Definition and Utilities*, in "Appendix: Converting DL/I Tables from IPF to ISPF Format."

If you do not want to use the DL/I IMF and IUG functions, consider the explanations under "Deleting the VSE Version of IMF and IUG" in topic 5.3.1 and "Deleting the VM Version of IMF and IUG" in topic 5.3.2.

If you plan to use the IMF and IUG functions in a:

° VSE environment, refer to "Using IMF and IUG Functions in a VSE Environment" in topic 5.3.3.

° VM environment, refer to "Using IMF and IUG Functions in a VM Environment" in topic 5.3.4.

Subtopics:

- 5.3.1 Deleting the VSE Version of IMF and IUG
- 5.3.2 Deleting the VM Version of IMF and IUG
- 5.3.3 Using IMF and IUG Functions in a VSE Environment
- 5.3.4 Using IMF and IUG Functions in a VM Environment

5.3.1 Deleting the VSE Version of IMF and IUG

If you do *not* plan to use the IMF and IUG functions under VSE, you may want to delete the IMF and IUG members from your DL/I libraries to save space. To do this, run the job stream shown in Figure 4. The job deletes all IMF and IUG phases, all IMF and IUG message files, menu panels, and skeletons, and all IMF and IUG object modules.

Note: The VSE version of DL/I IMF and IUG is not included with DL/I 1.10.

```
// JOB DELETVSE
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// EXEC LIBR,PARM='MSHP'
ACCESS SUBL=DLIPRD.sublib
DELETE DLZ0*.PHASE
DELETE DLZ0*.OBJ
DELETE *.M, *.N, *.S
/*
/&
```

Figure 4. Sample Job DELETVSE

5.3.2 Deleting the VM Version of IMF and IUG

If you do *not* plan to use the IMF and IUG functions under VM/CMS, you may want to delete the IMF and IUG members from your DL/I libraries to save space. To do this, run the job stream shown in Figure 5.

```
// JOB DELETVM
// DLBL DLIGEN,'DL/I-generation-library'
// EXTENT ,volid
// EXEC LIBR,PARM='MSHP'
ACCESS SUBLIB=DLIGEN.sublib
DELETE  DLZCMSTL.Z  DLZCMSML.Z
/*
/&
```

Figure 5. Sample Job DELETVM

5.3.3 Using IMF and IUG Functions in a VSE Environment

Note: The VSE version of DL/I IMF and IUG cannot be used under VSE/ESA.

If you plan to use the IMF or IUG functions under VSE, there are some steps necessary to set up the interface with ISPF. Before performing these steps, you must be familiar with VSE/ICCF and ISPF and with the following manuals:

- *VSE/Interactive Computing and Control Facility User's Guide*
- *Interactive System Productivity Facility (ISPF) Dialog Management Services*

Note: If you do not plan to use the IMF and IUG functions in a VM environment, you may want to delete the VM Version of IMF and IUG from your DL/I libraries to save space by running the job stream shown under "Deleting the VM Version of IMF and IUG" in topic 5.3.2, above.

Subtopics:

- 5.3.3.1 Accessing IMF and IUG Directly
- 5.3.3.2 Accessing IMF and IUG through an Existing ISPF Menu

5.3.3.1 Accessing IMF and IUG Directly

If you want to access the DL/I IMF and IUG functions directly, perform the following steps:

1. Get the ISPF startup command procedure, which is located in the ISPF sublibrary as member ISPSTART.I, and catalog it into an ICCF library.

2. Modify this procedure in ICCF:

   a. Add PANEL(DLZ0) as parameter to the ISPSTART command.

   b. Add the DL/I production sublibrary (which contains the IMF and IUG modules) to the search chain in the ISPDEF command.

3. Now, whenever you invoke the modified ISPF startup command procedure ISPSTART in ICCF, it will bring you into the DL/I master menu, directly.

5.3.3.2 Accessing IMF and IUG through an Existing ISPF Menu

If your ISPF menus do not contain a DL/I option, you must add this option by altering one of the ISPF menus. The following directions are valid for the ISPF master menu panel:

1. Get the ISPF master menu panel, member ISP@MSTR.N, from the ISPF sublibrary.

2. Modify this member as shown in the underlined lines of Figure 6:

   a. Add a new option to select DL/I (for example: D for DL/I) to the option part of the panel.

   b. Add another line for DL/I in the processing part (PROC). The new option used to select DL/I (D) should appear on the left side of the comma, and PANEL(DLZ0) should appear on the right side of the comma. It then would be: D,PANEL(DLZ0).

3. Add the appropriate job control statements as shown in Figure 6.

4. Recatalog the modified panel as member ISP@MSTR.N into the ISPF sublibrary by running the job in Figure 6.

5. Get the ISPF startup command procedure, which is located in the ISPF sublibrary as member ISPSTART.I, and catalog it into an ICCF library.

6. Modify this procedure in ICCF by adding the DL/I production sublibrary (which contains the IMF and IUG modules) to the search chain in the ISPDEF command.

7. Now, whenever you start up ISPF again, the ISPF master menu panel will contain DL/I as an additional option.

   Figure 6 shows the job to change the ISPF master menu panel (to allow accessing the DL/I IMF and IUG functions).
   The lines you have to change in the ISPF master menu panel are underlined (see Step 2, above).

```
// JOB ISPFCAT
// EXEC LIBR,PARM='MSHP'
ACCESS SUBLIB=ISPF.sublib
CATALOG ISP@MSTR.N REPLACE=YES
%----------------------  ISPF MASTER APPLICATION MENU  ---------------------
%OPTION  ===> ZCMD                                                        +
%                                                    +USERID   - &ZUSER
%   1 +SAMPLE1    - Sample application 1             +TIME     - &ZTIME
%   2 +.          - (Description for option 2)       +TERMINAL - &ZTERM
%   3 +.          - (Description for option 3)       +PF KEYS  - &ZKEYS
%   4 +.          - (Description for option 4)
%   D +DL/I       - DL/I Interactive Definitions
%   X +EXIT       - Terminate ISPF using list/log defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
  .HELP    = ISP00005     /* Help for this master menu             */
  &ZPRIM   = YES          /* This is a primary option menu         */
)PROC
```

```
    &ZSEL = TRANS( TRUNC (&ZCMD,'.')
                  1,'PANEL(ISP@PRIM)' /* Sample primary option menu  */
                  D,'PANEL(DLZ0)'     /* DL/I primary option menu    */
              /*******************************************************/
              /*                                                     */
              /* Add other applications here.                        */
              /*                                                     */
              /*******************************************************/
              /* Following shows how to code an invocation of the    */
              /* ISPF Program Development Facility, where "n" is     */
              /* the desired selection number:                       */
              /*                                                     */
              /*  n,'PANEL(ISR@PRIM) NEWAPPL(ISR)'                   */
              /*                                                     */
              /*******************************************************/
                ' ',' '
                X,'EXIT'
                *,'?' )
  )END
  /*
  /&
```

Figure 6. Sample Job ISPFCAT

5.3.4 Using IMF and IUG Functions in a VM Environment

If you plan to use the IMF and IUG functions under VM/CMS, you can install DL/I IMF and IUG functions onto VM for execution with ISPF. To use the functions, you need 4.23MB of disk space.

Subtopics:

- 5.3.4.1 Installing DL/I Dialogs on CMS

5.3.4.1 Installing DL/I Dialogs on CMS

To install the IMF and IUG members, perform the following steps.

1. Start up your VSE system and run the following VSE job:

```
// JOB CMS
// EXEC LIBR
  ACCESS SUBLIB=DLIGEN.sublib
  PUNCH DLZCMSTL.Z
  PUNCH DLZCMSML.Z
/*
/&
```

2. Route the punch output of the above job to a CMS userid, for example by issuing the CP spool command:

```
SPool PUNch userid
```

3. When the punch output of the VSE job is in your reader, receive it, for example by issuing the CMS command:

```
READCARD * * A
```

Ignore or erase the resulting CMS file:

```
READCARD CMSUT1 A
```

4. The output of this reading are the following VM files (they represent the VM version of IMF and IUG):

```
DLZ0TEXT TXTLIB A
DLZPANL1 MACLIB A
DLZPANL2 MACLIB A
DLZPANL3 MACLIB A
DLZSKELS MACLIB A
DLZMSGS  MACLIB A
```

5. Create and run a CMS EXEC containing all FILEDEFs for ISPF.

6. Enter ISPSTART PANEL(DLZ0) in CMS to access the IMF and IUG main menu.

For additional information, refer to the manual *DL/I Interactive Resource Definition and Utilities*.

5.4 DL/I Documentation Aid and the ISQL EXTRACT DEFINEs Utility

The DL/I Documentation Aid (DA) facility is an ease-of-use facility to document DL/I Data Base Description (DBD) and Program Specification Block (PSB) definitions that can be accessed directly by ISQL. This allows monitoring the DL/I database definitions. This facility is available to DL/I users who have SQL/DS***** and ISQL installed on their VSE system. When the Documentation Aid is invoked by the DL/I Application Control Blocks Creation and Maintenance utility, all the DBD and PSB definitions are automatically recorded into a special group of SQL/DS tables.

The ISQL EXTRACT DEFINEs utility provides the automatic creation of an ISQL routine of EXTRACT DEFINE commands, eliminating the repetitive task of identifying the DL/I databases to the ISQL EXTRACT facility. This utility obtains its definitions from the SQL/DS tables created by the DL/I Documentation Aid.

Before using the DL/I Documentation Aid facility and DL/I EXTRACT DEFINEs, you should be familiar with the following manuals:

- *DL/I Resource Definition and Utilities*, in
  - "Chapter: Defining a Program Specification Block"
  - "Chapter: Doing the ACBGEN Procedure"
- *SQL/Data System Administration for VSE*.

Note: Support of DL/I Extract has been removed by SQL/DS Version 3.

Subtopics:

- 5.4.1 DL/I Documentation Aid
- 5.4.2 ISQL EXTRACT DEFINEs

5.4.1 DL/I Documentation Aid

Listed below are the steps to install and activate the DL/I Documentation Aid (DA):

1. Create and run a job similar to job PUNCHDA, (Figure 7). This job punches three DL/I Documentation Aid job streams (provided in the production sublibrary) using the VSE/ESA librarian. Provide the proper job control information where indicated. The purpose of each job stream is:

DLZDATAB	Acquires DB space and creates the SQL/DS DL/I DA tables.
DLZDANDX	Creates the SQL/DS DL/I DA table indexes.
DLZDARTN	DATALOADs the ISQL DL/I DA routines in the routine table.

```
// JOB PUNCHDA
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// EXEC LIBR
ACCESS SUBL=DLIPRD.sublib
PUNCH DLZDATAB.A
PUNCH DLZDANDX.A
PUNCH DLZDARTN.A
/*
/&
```

Figure 7. Sample Job PUNCHDA. Use this job to acquire three DL/I Documentation Aid job streams from the production sublibrary.

2. Remove all the CATALOG and /+ statements from each of the three Documentation Aid job streams.

3. Ensure that the final two statements in each job stream are /* and /&.

4. Create two jobs similar to Figure 10 in topic 5.4.2, with READ MEMBER DLZDLBD and READ MEMBER DLZDLBP.

   Note: The modules DLZDLBD and DLZDLBP have been preprocessed by using SQL/DS Version 3 Release 2. If you are using a *different* release of SQL/DS, you must preprocess and reassemble these modules. To do this, create and run a job similar to that shown in Figure 9 in topic 5.4.2. This job stream creates the type A, type OBJ, and type PHASE entries for the modules.

   Figure 9 in topic 5.4.2 shows processing for module DLZDLBD. For the module DLZDLBP, change the names according to Figure 8 in topic 5.4.2.

5. All job control statements to be executed are set to invoke SQL/DS in single-partition mode.

   If you are running in:

   - Single-partition mode, update the EXEC PROC= statement to include your installation's procedure name for the SQL/DS database identification statements.

   - Multi-partition mode, delete the EXEC PROC= statement and change the EXEC statement for a multi-partition environment.

6. Update the password in all EXEC and SQL/DS CONNECT statements.

7. Remove the UPSI statement from the CRTDLBDP job if you want both punched and printed output.

8. Change the ACQUIRE statements in DLZDATAB only if it is necessary to increase your database space (DBSPACE).

9. Run the output (DLZDATAB, DLZDANDX, DLZDARTN) acquired in Step 1, plus the job shown in:

   - Figure 9 in topic 5.4.2, or
   - Figure 10 in topic 5.4.2.

10. After running all job streams, you can use the DL/I Documentation Aid.

5.4.2 ISQL EXTRACT DEFINEs

To use the ISQL EXTRACT facility, you need an SQL/DS prior to Version 3.

The ISQL EXTRACT module DLZEXDF has been preprocessed by using SQL/DS Version 2 Release 2. If you are using a different release of SQL/DS at your installation, you must preprocess and reassemble the module DLZEXDF. This must be done according to the naming convention shown in Figure 8 and using the sample job shown in Figure 9.

Module to be Preprocessed	Source to Catalog	Object Code to Catalog	Phase Name
DLZDLBD.A	DLZDLBDP.A	DLZDLBDP.OBJ	DLZDLBDP.PHASE
DLZDLBP.A	DLZDLBPP.A	DLZDLBPP.OBJ	DLZDLBPP.PHASE
DLZEXDF.A	DLZEXDFP.A	DLZEXDFP.OBJ	DLZEXDFP.PHASE

Figure 8. Naming Convention for DL/I DA and ISQL EXTRACT Modules

To install the ISQL EXTRACT DEFINEs utility, you have to submit a job for module DLZEXDF as shown in:

- Figure 9
  if your installed SQL/DS is *not* SQL/DS Version 2 Release 2.

- Figure 10
  if your installed SQL/DS *is* SQL/DS Version 2 Release 2.

Note that in Figure 9 and Figure 10:

- All job control statements are set to invoke SQL/DS in single-partition mode.

  If you are running in:

  - Single-partition mode, update the EXEC PROC= statement to include your installation's procedure name for the SQL/DS database identification statements.

  - Multi-partition mode, delete the EXEC PROC= statement and change the EXEC statement for a multi-partition environment.

- You have to update the password in the EXEC statement.

To continue the DL/I EXTRACT DEFINEs process, perform the following steps:

1. Start the ISQL EXTRACT DEFINEs utility using the following EXTRACT parameter statement:

```
EXTRACT
   PCBNAME=pcbname,
   PSBNAME={psbname|(psbname,pcbnumber)},
   DLIPROC=procname,
   USERID=SQLDBA/password
   [,REPLACE]
```

2. Run the ISQL routine identified by the PCBNAME parameter specified in the above EXTRACT statement.

3. Complete the ISQL processing by:

   Creating the Target Tables
   Issuing the EXTRACT command
   Issuing the SUBMIT command

```
// JOB PREP DLZDLBD
// LIBDEF *,SEARCH=(SQL.sublib,DLIPRD.sublib,DLIGEN.sublib,...)
// LIBDEF PHASE,CATALOG=DLIPRD.sublib
// EXEC PROC=(identification statements for sql/ds database)
// DLBL IJSYSPH,'punch-work-file',0
// EXTENT SYSPCH,,1,0,xxx,yyy
// DLBL IJSYSIN,'punch-work-file',0
// EXTENT SYSIPT,,1,0,xxx,yyy
ASSGN SYSPCH,DISK,VOL=volid,SHR
* ---------------------------------------------------------- *
* STEP1:                                                     *
*       PREPROCESS DLZDLBD                                   *
* ---------------------------------------------------------- *
// EXEC PGM=ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,PROGNAME=ARIPRPA/PREPNAME=DLZDLBDP,
        USERID=sqldba/sqldbapw'
READ MEMBER DLZDLBD
/+
/*
CLOSE SYSPCH,PUNCH
* ---------------------------------------------------------- *
* STEP2:                                                     *
*       CATALOG PREPROCESSED SOURCE TO DLZDLBDP.SQL          *
*       NOTE: FOR LIBRARIAN PURPOSES THE SOURCE MUST NOT     *
*             BE OF TYPE "A"                                 *
* ---------------------------------------------------------- *
ASSGN SYSIPT,DISK,VOL=volid,SHR
// EXEC LIBR,PARM='ACCESS S=DLIGEN.sublib; CATALOG DLZDLBDP.SQL REPLACE=Y'
/*
CLOSE SYSIPT,SYSRDR
* ---------------------------------------------------------- *
* STEP3:                                                     *
*       REMOVE OLD DLZDLBDP.A                                *
*       RENAME DLZDLBDP.SQL ----> DLZDLBDP.A                 *
* ---------------------------------------------------------- *
// EXEC LIBR,PARM='MSHP'
ACCESS S=DLIGEN.sublib
DEL DLZDLBDP.A
REN DLZDLBDP.SQL : DLZDLBDP.A
/*
* ---------------------------------------------------------- *
* STEP4:                                                     *
*       ASSEMBLE  DLZDLBDP.A                                 *
* ---------------------------------------------------------- *
ASSGN SYSPCH,DISK,VOL=volid,SHR
// OPTION CATAL,DECK,ALIGN
// EXEC ASSEMBLY,SIZE=1024K
         COPY   DLZDLBDP
/*
CLOSE SYSPCH,PUNCH
* ---------------------------------------------------------- *
* STEP5:                                                     *
*       CATALOG  DLZDLBDP.OBJ                                *
* -------------------------------------------------------- *
ASSGN SYSIPT,DISK,VOL=volid,SHR
// EXEC LIBR,PARM='ACCESS S=DLIPRD.sublib'
/*
CLOSE SYSIPT,SYSRDR
* -------------------------------------------------------- *
* STEP6:                                                   *
*       LINKEDIT DLZDLBDP.OBJ                              *
* -------------------------------------------------------- *
// OPTION CATAL
  INCLUDE DLZDLBDP
// EXEC LNKEDT
/*
/&
```

Figure 9. Sample Job PREP DLZDLBD

```
// JOB CRTDLBDP
// UPSI 11
// LIBDEF *,SEARCH=(SQL.sublib,DLIPRD.sublib,DLIGEN.sublib,...)
// EXEC PROC=(identification statements for sql/ds database)
// EXEC PGM=ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,PROGNAME=ARIPRPA/PREPNAM*
               E=DLZDLBDP,USERID=sqldba/sqldbapw'
READ MEMBER DLZDLBD
/*
/&
```

Figure 10. Sample Job CRTDLBDP

6.0 Chapter 6. Post Installation Activities and Aids

Subtopics:

- 6.1 How to Get the DL/I Sample Application Job Streams
- 6.2 MSHP Retrace
- 6.3 Displaying DL/I Library Directory Listings
- 6.4 DL/I Link Book
- 6.5 DL/I Merge Book
- 6.6 DL/I Delete Book
- 6.7 Debugging Aids

6.1 How to Get the DL/I Sample Application Job Streams

To verify the correct installation of DL/I, you can use the two sample application job streams that are described in the manual *DL/I Guide for New Users*, in "Chapter: DL/I Sample Application."

1. The *Online* sample application job stream is in the DL/I production sublibrary as member DLZSAMJS.A. Run the following sample job to get this member from the DL/I production sublibrary:

```
// JOB PUNCH
// EXEC LIBR
  ACCESS SUBLIB=DLIPRD.sublib
  PUNCH DLZSAMJS.A
/*
/&
```

2. The *Access* sample application job stream is in the DL/I production sublibrary as member DLZSAMAC.A.

You can retrieve all other DL/I sample programs in the same way (refer to "Source Books in the Production Sublibrary" in topic 3.1.1).

6.2 MSHP Retrace

After you have installed DL/I, you may wish to print the DL/I related information contained in the System History File.

1. *IBM Service* panel of the VSE Interactive User Interface

   6 (Retrace History File), and then

   7 (Retrace Component ID)

2. Run a job similar to the DL/I 1.10 sample job shown below:

```
// JOB RETRACE
// EXEC MSHP,SIZE=900K
     RETRACE COMP ID=5746-XX1-00-DB5
/*
/&
```

6.3 Displaying DL/I Library Directory Listings

To display the directory listings of your DL/I sublibraries, run the jobs in Figure 11 and Figure 12:

```
// JOB DSPLYPRD
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// EXEC LIBR
LISTD SUBL=DLIPRD.sublib
/*
/&
```

Figure 11. Sample Job DSPLYPRD.
Use this job to display the directory of the DL/I production sublibrary.

```
// JOB DSPLYGEN
// DLBL DLIGEN,'DL/I-generation-library'
// EXTENT ,volid
// EXEC LIBR
LISTD SUBL=DLIGEN.sublib
/*
/&
```

Figure 12. Sample Job DSPLYGEN.
Use this job to display the directory of the DL/I generation sublibrary. Do not run this job, if you did not install the generation sublibrary.

6.4 DL/I Link Book

If you change any of the DL/I source modules, the affected DL/I production sublibrary phases must be relinked. Member DLZLNKBK.A in the DL/I production sublibrary contains job control statements to link all DL/I modules. You can punch out this link edit book using the VSE librarian program with the PUNCH command. Then, add your own job control statements to the member obtained from the library.

Linking certain DL/I phases may produce various linkage editor messages. For more information about these messages, refer to "Error Messages while Reassembling or Relinking DL/I Components" in topic 7.4.

6.5 DL/I Merge Book

If you wish to merge any or all of the DL/I code from the DL/I libraries to any other libraries, you can use the merge book (DLZMERGE.A) that is included in the DL/I production sublibrary. You can punch out this merge book using the VSE librarian program with the PUNCH command. Then, read the comments, alter the statements provided, add the necessary job control statements and run the job by using the VSE librarian program.

6.6 DL/I Delete Book

If you wish to delete any or all DL/I code from the libraries where your DL/I code resides, you can use the delete book (DLZDLETE.A) that is provided in the DL/I production sublibrary. You can punch out this delete book using the VSE librarian program with the PUNCH command. Then, read the comments, alter the statements provided, add the necessary job control statements and run the job by using the VSE librarian program.

6.7 Debugging Aids

To assist in problem determination and solution, DL/I produces three types of dumps:

Unformatted Dumps
Display register contents and the contents of a section of storage at the time of the failure.

Formatted Dumps
Provide additional information by means of:

- Locating DL/I control blocks in storage
- Dumping each control block separately
- Identifying each block with a control block heading

Problem Determination Dumps
Are unformatted dumps that are identified by a special header and written to a special dump data set.

DL/I dumps are taken in the event of the following:

- Online task failure
- Online system failure
- Batch application failure
- MPS application failure
- Batch utility failure (where the utility is using DL/I)

where *failure* is an abnormal ending of a job or task.

The following table shows what type of dump is useful when analyzing one of the above failures:

Type of Failure	Unformatted Dump	Formatted Task Dump	Formatted System Dump	Problem Determination Dump
Online Task		x		
Online System			x	x
Batch Application			x (1)	x
MPS application	x			x
Batch utility	x		x	x

(1) Only if // OPTION NOSYSDMP specified in JCL.

For additional information about DL/I debugging facilities and dump selection, refer to the manual *DL/I Diagnostic Guide*, in "Chapter: Interpreting And Debugging Dumps."

7.0 Chapter 7. Applying Service

Subtopics:

- 7.1 Reassembling CICS Modules for Updated DL/I
- 7.2 Reassembling and Relinking DL/I for Updated CICS
- 7.3 Service for DL/I Application Interface Modules
- 7.4 Error Messages while Reassembling or Relinking DL/I Components

7.1 Reassembling CICS Modules for Updated DL/I

There is a set of CICS* modules that must be reassembled whenever significant service is applied to DL/I (as indicated in the PTF cover letter).

The CICS system generation macro instructions for this have been provided in member DFHSGDLI.J as part of the CICS generation library. (The CICS generation sublibrary is part of the VSE generation library.)

If you want to reassemble the necessary CICS modules, you must:

1. Get member DFHSGDLI.J from the CICS generation sublibrary by running a VSE librarian job similar to the following example:

```
 ________________________________________________________________________
|                                                                        |
|                                                                        |
|      // JOB PUNCH                                                      |
|      // DLBL CICSLB,'CICS-library'                                     |
|      // EXTENT ,volid                                                  |
|      // EXEC LIBR                                                      |
|      ACCESS SUBLIB=CICSLB.sublib                                       |
|      PUNCH DFHSGDLI.J                                                  |
|      /*                                                                |
|      /&                                                                |
|                                                                        |
|                                                                        |
|________________________________________________________________________|
```

Figure 13. Sample Job PUNCH.
Use this job to acquire the CICS job stream in member DFHSGDLI.J from the CICS generation sublibrary.

2. Remove all the CATALOG and /+ statements from the job stream and ensure that the final two statements in the job are /* and /&.

3. Add the DLBL, EXTENT and the LIBDEF job control statements as shown in the sample job stream CICSGEN in Figure 14.

4. Run this job.

   The output of job CICSGEN will be a file containing 7 jobs in your punch queue. You can run these jobs together or individually.

   Notes:

   a. The assemblies of the CICS modules can take a significant amount of execution time.

   b. **Unresolved External References**:
      When you run the assembly and link edit jobs for the DBP and DLBP programs, you will see two unresolved external references messages, for DLZPRNT and DLZSLOG, in the link edit map. These are not important and you can safely ignore them.

.

```
  // JOB CICSGEN
  // DLBL CICSLB,'CICS-library'
  // EXTENT ,volid
  // DLBL DLIPRD,'DL/I-production-library'
  // EXTENT ,volid
  // DLBL VSAMLB,'VSAM-library'
  // EXTENT ,volid
  // LIBDEF *,SEARCH=(CICSLB.sublib,DLIPRD.sublib,VSAMLB.sublib)
  // OPTION DECK,ALIGN
  // EXEC ASSEMBLY
           DFHSG TYPE=INITIAL,                                          X
                 OPSYS='VSE',                                           X
```

```
                  ASMBLR=ASSEMBLY,                                      X
                  PRINT=(LIST,XREF,NODSECT),                            X
                  STAGE2=FORCE,                                         X
                  EJECT=YES,                                            X
                  JOBNAME=DFH,                                          X
                  DLI=YES,                           DL/I Support       X
                  STARTER=YES,                                          X
                  MOD=(DBP,2$,DLBP,,ISP,,RUP,,JCP,,XFP,),               X
                  ACCTID='CICS_220'
            DFHSG PROGRAM=PREGEN
            DFHSG TYPE=FINAL
            END
   /*
   /&
```

Figure 14. Sample Job CICSGEN.
Use this job stream to create the necessary jobs to reassemble the following six **CICS/VSE* 2.2** modules:
DFHDBP2$, DFHDLBP, DFHISP, DFHRUP, DFHJCP, DFHXFP
It will also create one additional job to print a listing of the DSECTs used.
Note that DFHXFP and DFHISP are SVA eligible and may have to be replaced in the SVA.

7.2 Reassembling and Relinking DL/I for Updated CICS

Whenever a new CICS that is later than:

```
CICS/VSE* 2.2 with DL/I 1.10
CICS/VSE* 2.1 with DL/I 1.9
CICS/DOS/VS 1.7 with DL/I 1.8
```

is installed, or if significant CICS service is applied, the steps listed below have to be carried out.

Note: If you are using VSE/Advanced Functions 2.1 (and you are not a VSE/SP user), you must define the VSE/Advanced Functions generation sublibrary in the LIBDEF chain before performing these steps.

1. Edit the following DL/I macro (to include possibly updated CICS macros) by running sample job ASSEM1 in Figure 15:

DLZPRECC	Preparation for CICS Journal

2. Assemble, catalog, and link edit the following DL/I modules (which are using CICS) by running sample job ASSEM2 in Figure 16 for each of these modules. The source of these modules is in the DL/I production sublibrary as type A members:

DLZBPC00	MPS Batch Partition Controller
DLZMPC00	MPS Master Partition Controller

DLZMPI00	MPS Batch Interface
DLZMPUR0	MPS Purge TSQ Transaction
DLZMSTP0	MPS Stop Transaction
DLZMSTR0	MPS Start Transaction
DLZOLI00	DL/I Online Initialization
DLZRDBL1	DL/I Online Journaling
DLZSTP00	DL/I System Termination
DLZSTTL	Run and Buffer Statistics (Online)

3. Assemble and catalog the following DL/I modules (that are using CICS) by running sample job ASSEM3 in Figure 17. The source code of these modules is in the DL/I production sublibrary as type A members:

DLZEIPO0	HLPI EXEC Interface Program (Online)
DLZFTDP0	DL/I Formatted Task Dump
DLZISC00	Intersystems Communication
DLZLOC00	Local PSB Scheduling
DLZODP	DL/I Online Interface
DLZSTRO0	Field Storage Manager (Online)

4. Reassemble DL/I ACTs using the DLZACT macro and relink the ACTs.

5. Edit the following DL/I macro (to include possibly updated CICS macros) by running sample job ASSEM4 in Figure 18:

DLZTRACE	Trace Program

6. Reassemble and relink all your trace programs that have the option OUTPUT=CICS.

```
// JOB ASSEM1
// OPTION DECK,NOXREF,NOEDECK
// DLBL CICSLB,'CICS-library'
// EXTENT ,volid
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// LIBDEF *,SEARCH=(CICSLB.sublib,DLIPRD.sublib,other-sublibs)
// DLBL IJSYSPH,'punch-work-file',0
// EXTENT SYSPCH,,1,0,xxx,yyy
ASSGN SYSPCH,DISK,VOL=volid,SHR
// EXEC ASSEMBLY
        PUNCH 'ACCESS SUBLIB=DLIPRD.sublib'
        END
/*
// OPTION EDECK,NOXREF,NODECK
// EXEC ASSEMBLY
        PRINT  NOGEN
        COPY DLZPRECC
        END
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'punch-work-file',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// EXEC LIBR,PARM='MSHP'
/*
CLOSE SYSIPT,reader
```

```
   /&
```

Figure 15. Sample Job Stream ASSEM1.
Use this job stream to assemble the DLZPRECC macro (member of type A) and to catalog the output as DLZPRECC.E into your DL/I production sublibrary.

```
   // JOB ASSEM2
   // OPTION DECK,NOXREF,NOEDECK
   // DLBL CICSLB,'CICS-library'
   // EXTENT ,volid
   // DLBL DLIPRD,'DL/I-production-library'
   // EXTENT ,volid
   // LIBDEF *,SEARCH=(CICSLB.sublib,DLIPRD.sublib,other-sublibs)
   // LIBDEF PHASE,CATALOG=DLIPRD.sublib
   // DLBL IJSYSPH,'punch-work-file',0
   // EXTENT SYSPCH,,1,0,xxx,yyy
   ASSGN SYSPCH,DISK,VOL=volid,SHR
   // EXEC ASSEMBLY
           PUNCH 'ACCESS SUBLIB=DLIPRD.sublib'
           END
   /*
   // EXEC ASSEMBLY
           PRINT NOGEN
           COPY DLZxxxxx
           END
   /*
   CLOSE SYSPCH,punch
   // DLBL IJSYSIN,'punch-work-file',0
   // EXTENT SYSIPT
   ASSGN SYSIPT,DISK,VOL=volid,SHR
   // EXEC LIBR,PARM='MSHP'
   /*
   CLOSE SYSIPT,reader
   // OPTION CATAL
      INCLUDE DLZxxxxx
   // EXEC LNKEDT,PARM='MSHP'
   /&
```

Figure 16. Sample Job Stream ASSEM2.
Use this job to reassemble, catalog and link edit the following 10 DL/I modules (members of type A):

DLZBPC00	DLZMPC00	DLZMPUR0	DLZMSTP0	DLZMSTR0
DLZOLI00	DLZRDBL1	DLZSTP00	DLZSTTL	DLZMPI00

Each of these modules requires its own separate job stream (that is: one complete job stream for each DL/I module). Substitute the respective module name for DLZ*xxxxx* in the COPY and INCLUDE statement.

```
   // JOB ASSEM3
   // OPTION DECK,NOXREF,NOEDECK
   // DLBL CICSLB,'CICS-library'
   // EXTENT ,volid
   // DLBL DLIPRD,'DL/I-production-library'
   // EXTENT ,volid
```

```
// LIBDEF *,SEARCH=(CICSLB.sublib,DLIPRD.sublib,other-sublibs)
// DLBL IJSYSPH,'punch-work-file',0
// EXTENT SYSPCH,,1,0,xxx,yyy
ASSGN SYSPCH,DISK,VOL=volid,SHR
// EXEC ASSEMBLY
        PUNCH 'ACCESS SUBLIB=DLIPRD.sublib'
        END
/*
// EXEC ASSEMBLY
        PRINT NOGEN
        COPY DLZxxxxx
        END
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'punch-work-file',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// EXEC LIBR,PARM='MSHP'
/*
CLOSE SYSIPT,reader
/&
```

Figure 17. Sample Job Stream ASSEM3.
Use this job to assemble and recatalog the following 6 DL/I modules (members of type A):
DLZEIPO0   DLZFTDP0   DLZISC00   DLZLOC00   DLZODP DLZSTRO0
Each of these modules requires its own separate job stream (that is: one complete job stream for each DL/I module).
Substitute the respective module name for DLZ*xxxxx* in the COPY statement.

```
// JOB ASSEM4
// OPTION DECK,NOXREF,NOEDECK
// DLBL CICSLB,'CICS-library'
// EXTENT ,volid
// DLBL DLIPRD,'DL/I-production-library'
// EXTENT ,volid
// LIBDEF *,SEARCH=(CICSLB.sublib,DLIPRD.sublib,other-sublibs)
// DLBL IJSYSPH,'punch-work-file',0
// EXTENT SYSPCH,,1,0,xxx,yyy
ASSGN SYSPCH,DISK,VOL=volid,SHR
// EXEC ASSEMBLY
        PUNCH 'ACCESS SUBLIB=DLIPRD.sublib'
        END
/*
// OPTION EDECK,NOXREF,NODECK
// EXEC ASSEMBLY
        PRINT NOGEN
        COPY DLZTRACE
        END
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'punch-work-file',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// EXEC LIBR,PARM='MSHP'
/*
CLOSE SYSIPT,reader
/&
```

```
|                                                                       |
|_______________________________________________________________________|
```

Figure 18. Sample Job Stream ASSEM4. ***Only online users should use this job.***
Run this job, if you have TRACE programs with OUTPUT=CICS. Sample job ASSEM4 assembles the DLZTRACE macro (member DLZTRACE.A) and catalogs it as member DLZTRACE.E into the DL/I production sublibrary for use in later assemblies of your TRACE programs.
After running this job, you must reassemble and relink all of your TRACE programs that have the option OUTPUT=CICS.

7.3 Service for DL/I Application Interface Modules

To ease migration for DL/I applications from 24-bit to 31-bit mode, DL/I 1.10 delivers its application interface modules:

- DLZBPJRA
- DLZLI000
- DLZLICBL

with the attributes **AMODE=ANY** and **RMODE=ANY**.

This allows you, for example, to save updates of link books for DL/I applications with the linkage editor MODE control statement.

For applying service to the DL/I application interface modules, note the following:

- If you want to apply service by means of Authorized Program Analysis Report (APAR) fixes, the High Level Assembler must be used to reassemble the modules. Refer to Figure 19.

- If the High Level Assembler is not available, the modules have to be serviced by means of Program Temporary Fix (PTF).

```
 _______________________________________________________________________
|                                                                       |
|                                                                       |
|   // JOB HLASM                                                        |
|   // OPTION DECK                                                      |
|   // DLBL DLIPRD,'DL/I-production-library'                            |
|   // EXTENT ,volid                                                    |
|   // DLBL HLALIB,'High-level-assembler-library'                       |
|   // EXTENT ,volid                                                    |
|   // LIBDEF *,SEARCH=(DLIPRD.sublib,HLALIB.sublib)                    |
|   // DLBL IJSYSPH,'punch-work-file',0                                 |
|   // EXTENT SYSPCH,,1,0,xxx,yyy                                       |
|   ASSGN SYSPCH,DISK,VOL=volid,SHR                                     |
|   // EXEC ASMA90,SIZE=...                                             |
|            PUNCH 'ACCESS SUBLIB=DLIPRD.sublib'                        |
|            COPY DLZBPJRA                                              |
|            END                                                        |
|   /*                                                                  |
|   CLOSE SYSPCH,punch                                                  |
|   // DLBL IJSYSIN,'punch-work-file',0                                        |
|   // EXTENT SYSIPT                                                           |
|   ASSGN SYSIPT,DISK,VOL=volid,SHR                                            |
|   // EXEC LIBR,PARM='MSHP'                                                   |
|   /*                                                                         |
|   CLOSE SYSIPT,reader                                                        |
|   /&                                                                         |
|                                                                              |
|                                                                              |
|______________________________________________________________________________|
```

Figure 19. Sample Job Stream HLASM. Run this job, if APAR service is to be applied to one of the DL/I application interface modules using the High Level Assembler.

7.4 Error Messages while Reassembling or Relinking DL/I Components

If you intend to assemble any of the DL/I module source books from the DL/I libraries, certain assembly error or warning messages may result. Such messages do not affect the assembler output in any way. Described below are some of the assembler messages you may encounter:

° IPK001 END STATEMENT IN MACRO OR COPY CODE

  This warning message may be removed by deleting the END statement from the DL/I source book, prior to the assembly. If the END statement which was removed contains an ENTRY POINT ADDRESS, that ENTRY POINT must be included in the END statement in the assembly job stream.

° IPK182 ALIGNMENT ERROR IN OPERAND 2

  This warning message may be issued during the assembly of some DL/I modules, such as DLZDLOC0, DLZTPRT0, DLZFSDP0, or DLZTRPR0. The message can be ignored and the assembly is complete.

° IPK202 TITLE NAME TOO LONG

  This warning message indicates that the module title statement is too long. The assembly continues and the message may be ignored.

In addition, when relinking certain DL/I modules and creating DL/I phases for the production sublibrary, certain warning messages may appear on the console and in the output (SYSLST) of the link edit job. Described below are some of the warning link edit messages you may encounter on your console log and/or on SYSLST (output listing):

° 2158I   NO CSECT LENGTH SUPPLIED

  The END statement does not contain the length of the control section. Allow the job to execute. If execution fails, reassemble (recompile) and relink the module in question. If execution is successful, ignore the message. This may appear when link editing phases such as DLZMPI00, DLZRRC00, DLZBPC00.

- 2199I ERROR HAS OCCURRED DURING LINKAGE EDITING

  The specific messages will appear on SYSLST and processing will continue. Review the messages, take any action specified and continue running the link edit job.

- 2139I DUPLICATE SECTION DEFINITION _____ **** SECTION IGNORED ****

  This may appear when link editing a number of DL/I phases and the associated modules and CSECTs included. The link edit should continue to process, then review the output (SYSLST) for any additional information or action required.

  Listed below are the DL/I phases where this condition may occur:

```
     DLZBACK0,  DLZBPC00,  DLZBNUC0,  DLZDBH00,  DLZDHDS0,
     DLZDLA00,  DLZDLBDP,  DLZDLBPP,  DLZDLR00,  DLZDSEH0,
     DLZEXDFP,  DLZFSDP0,  DLZLOGP0,  DLZMPI00,  DLZRDBL0,
     DLZRRC00,  DLZTPRT0,  DLZUACB0,  DLZUCUM0,  DLZUDMP0,
     DLZURDB0,  DLZURGL0,  DLZURGP0,  DLZURGS0,  DLZURGU0,
     DLZURG10,  DLZURPR0,  DLZURRL0,  DLZURUL0.
```

  In addition the following CSECTs (duplicate section names) may also appear in the above message when link editing certain DL/I phases:

```
     DLZCONSL,  DLZDIMOD,  DLZLGPCN,  DLZLGPMT,  DLZRDR,
     DLZRRC10,  DLZTPCN,   IJJFCBIC,  OPENWORK,  SCDCSECT.
```

- CONTROL SECTIONS OF ZERO LENGTH IN INPUT

  This may appear when link editing such phases as:

```
     DLZBNUC0,  DLZDDLE0,  DLZDHDS0,  DLZDLRA0,  DLZMDLI0,
     DLZMPI00,  DLZMPUR0,  DLZMSTP0,  DLZMSTR0,  DLZOLI00,
     DLZRRC00,  DLZUACB0.
```

  The link edit should continue and you may ignore the message and continue processing.

- UNRESOLVED EXTERNAL REFERENCES

  This may appear when link editing phases DLZDLBDP, DLZDLBPP, and DLZEXDFP. The link edit should continue to execute and the message may be ignored.

- POSSIBLE INVALID ENTRY POINT DUPLICATION IN INPUT

  This may occur in some phases such as DLZDHSD0 and DLZUDMP0. The link edit should continue processing and you may ignore the message.

- UNRESOLVED ADDRESS CONSTANTS

This is a valid condition when link editing a number of DL/I phases such as:

- DLZURG10, DLZBACK0 (with CICS), DLZDBH00.

- DLZACT (application control table, which is the method of link editing the DL/I online nucleus: DLZODP, DLZNUC*xx*, and DLZNUC).

- DLZRDBC0 (when linked with DFHDBP and DFHDLBP).

The condition also occurs when link editing the modules DLZDLBDP, DLZDLBPP, and DLZEXDFP.

The unresolved external references are caused by weak external references (WXTRNs) which cause no operational error. Check the output (SYSLST) and continue the link edit processing.

In most of the cases explained above, the assembler and/or link edit process continues successfully. However, you should review any warning or error messages which appear during reassembling or relinking. Refer to the appropriate *VSE/SP Messages and Codes* manual, and take the necessary actions indicated.

8.0 Chapter 8. New Features

Subtopics:

- 8.1 New Features in DL/I 1.10
- 8.2 New Features in DL/I 1.9
- 8.3 New Features in DL/I 1.8
- 8.4 New Features in DL/I 1.7.1

8.1 New Features in DL/I 1.10

DL/I 1.10 exploits the capability of *31-bit addressing* that is available with VSE/ESA* 1.3 for:

◦ **DL/I Applications above the 16MB Line of Storage**

For detailed information, refer to "DL/I Applications above the 16MB Line of Storage" in topic 9.1.

◦ **VSE/VSAM HS-Buffers above the 16MB Line of Storage**

For detailed information, refer to "VSE/VSAM HS-Buffers above the 16MB Line of Storage" in topic 9.2.

Also, DL/I 1.10 exploits the *VSE/ESA virtual disk support*. That is, work files may reside in data space instead of residing on real devices. For detailed information, refer to "Virtual Disk Exploitation" in topic 9.3.

Also, changes have been made to the DL/I statistics program DLZSTTL. Using the program, you now can get expanded and further information about the DL/I calls, the subpools, and the various data bases.
For detailed information, refer to "DL/I Run Statistics (DLZSTTL) Enhancements" in topic 9.4.

8.2 New Features in DL/I 1.9

Subtopics:

- 8.2.1 Support of VSE/ESA

8.2.1 Support of VSE/ESA

DL/I 1.9 supports the VSE/ESA dynamic partitions, a CICS/VSE***** 2.1 XRF environment and an enhanced database I/O error handling. For details refer to Chapter 10, "DL/I 1.9 Support for VSE/ESA 1.1" in topic 10.0.

8.3 New Features in DL/I 1.8

Subtopics:

- 8.3.1 Date and Time on Reports and Statistics
- 8.3.2 Literal String in the HLPI WHERE Clause
- 8.3.3 Conditional Job Control Support
- 8.3.4 Status Code for Read-Only Programs
- 8.3.5 Partial Data Base Load
- 8.3.6 Increased Maximum Data Set Control Interval Size
- 8.3.7 Performance Improvements for Data Set Image Copy/Recovery
- 8.3.8 Device Independence
- 8.3.9 Automatic Verification for DL/I ESDS Data sets
- 8.3.10 Program Isolation Enhancement

8.3.1 Date and Time on Reports and Statistics

Reports and statistics provided by several DL/I utilities are date and time stamped and have headings of a unique form.

------------------ General-use programming interface -------------------

8.3.2 Literal String in the HLPI WHERE Clause

The DL/I High Level Programming Interface (HLPI) is an easy-to-use method for processing DL/I databases. It provides commands similar in syntax to those in CICS command language.

The existing syntax has been extended to allow the use of literals with the WHERE option in PL/I programs. With this support, the WHERE option will be compatible with the OS EXEC DLI language.

The method for selecting a segment with the WHERE clause using HLPI for PL/I programs is as follows:

```
     WHERE(name op value)
```

where:

name is the name of any field defined in the segment.

op is a relational operator.

value is a comparative value, which is:

- a variable name declared in the host language of the application program, or
- a character string for PL/I programs. The literal must be enclosed in quotes.

The HLPI support is dependent on the CICS EXEC translator support.

Within syntax checking by the CICS EXEC translator, the following errors are detected by the translator:

- Invalid text in the WHERE clause
- Missing quote

Diagnostic messages issued by the CICS EXEC translator are described in the manual *VSE/ESA Messages and Codes*.

|--------------- End of General-use programming interface ---------------|

8.3.3 Conditional Job Control Support

DL/I supports the use of conditional JCL for VSE. The DL/I batch initialization program DLZRRC00, the MPS batch initialization program DLZMPI00, and the DL/I utilities always provide return codes. Return codes provided by user-written programs are passed on by request.

For details, refer to the manual *DL/I Resource Definition and Utilities*.

8.3.4 Status Code for Read-Only Programs

A new processing option, PROCOPT, is provided for read-only programs to give the application a possibility to react on invalid pointer situations and to reduce the number of abnormal terminations.

For details refer to the manual *DL/I Resource Definition and Utilities*.

8.3.5 Partial Data Base Load

This feature allows initial loading of a database with a large amount of data in more than one step, using PROCOPT=L. Read-only access within the several load steps is possible.

For details refer to the manual *DL/I Resource Definition and Utilities*.

8.3.6 Increased Maximum Data Set Control Interval Size

DL/I 1.8 supports VSAM CI-sizes of its databases up to a maximum of 30KB.

For details refer to the manual *DL/I Resource Definition and Utilities*.

8.3.7 Performance Improvements for Data Set Image Copy/Recovery

A block size for DL/I backup tapes can optionally be specified.

For details refer to the manual *DL/I Resource Definition and Utilities*.

8.3.8 Device Independence

This feature is available for all users of DL/I 1.8 with APAR PL48345.

Due to a new CKD specification, DL/I database definition is device-independent for CKD devices.
If you use the DATASET parameter during the database definition, you may code a new specification of the DEVICE parameter:

```
    DEVICE=CKD  Must only be specified for CKD devices.
    DEVICE=FBA  Must only be specified for FBA devices.
```

Coding Example:

```
     DATASET DEVICE=CKD ...
```

The CKD device type specifications (for example, 3380) need no longer be used but are still supported. The new device specification is also supported by the Interactive Macro Facility (IMF).

For new messages refer to Chapter 11, "New Messages" in topic 11.0.

8.3.9 Automatic Verification for DL/I ESDS Data sets

This feature is available for all users of DL/I 1.8 with APAR PL20988.

Automatic Verification for DL/I ESDS data sets allows you to have just one CICS-DL/I startup with no VERIFY-statements in it, independent of a normal or an emergency start.
VSE/VSAM 1.3 provides automatic verification for all VSAM data sets doing normal record processing. Thus, the index component of a DL/I database will be verified automatically. Automatic verification does now also apply to a data component of a DL/I database (ESDS) working in Control Interval Processing Mode (CNV).

8.3.10 Program Isolation Enhancement

This feature is available for all users of DL/I 1.8 with APAR PL55587.

Due to the DL/I program isolation enhancement, a long running transaction may no longer cause a large amount of storage to be acquired and not freed.

8.4 New Features in DL/I 1.7.1

The features described in this chapter are available for all users of DL/I 1.7.1 and DL/I 1.7.0 users with SPE PTF UL90011:

- With VSE/Advanced Functions 2.1 (part of VSE/SP 2.1 and VSE/SP 3.1), DL/I can make full use of the VSE Virtual Addressability Extension with Multiple Partition Support (MPS).
- The new GSTA call can be used to acquire system statistics.

Subtopics:

- 8.4.1 Improved Multiple Partition Support
- 8.4.2 Dynamically Scheduling MPS or Non-MPS Execution
- 8.4.3 The New GSTA Call
- 8.4.4 GSTA Call Format
- 8.4.5 GSTA Return Codes
- 8.4.6 Layout of the Statistics Buffer

8.4.1 Improved Multiple Partition Support

DL/I supports separate address spaces for MPS partitions. However, CICS and DL/I must still be in the same partition. With the exception of applications using GSCD to get buffer statistics or to reference other DL/I control blocks, the support is transparent. No special procedures are required by DL/I for initialization of a DL/I MPS environment in multiple address spaces.

8.4.2 Dynamically Scheduling MPS or Non-MPS Execution

If you want to run a DL/I batch job in an MPS batch partition and you are not sure if MPS is already started, you can use the sample program DLZCTRL (shown in Figure 20). DLZCTRL will fetch either:

```
DLZRRC00 (if MPS is not active), or
DLZMPI00 (if MPS is already started).
```

By specifying DLZCTRL on the EXEC statement, you can dynamically schedule MPS or non-MPS execution.

Note: The sample program DLZCTRL shown in Figure 20, is not included on the distribution tape of DL/I 1.7.1.

This sample program does for DL/I 1.7.1 what sample program DLZCTRL does for DL/I 1.7.0. For information on the sample program DLZCTRL, refer to the manual *DL/I Guide for New Users*, in "Appendix A: DL/I Initialization" under "Dynamically Scheduling MPS or Non-MPS Execution."

DLZCTRL uses the XPCC FUNC=IDENT macro to test for the presence of the start partition XPCCB of MPS (APPL=SYSDLIO1, TOAPPL=SYSDLIB1):

° If this XPCC communication link in MPS online is currently set up (R15&nesym.0 and return code IJBXRETC reflects IDENT already done), then MPS is active and the program fetches the phase DLZMPI00.

° If MPS is not active, then the program fetches the phase DLZRRC00.

This technique will not work if logging is required, because there is no way to dynamically assign the DL/I log device if non-MPS operation is required.

```
  DLZCTRL  CSECT
           BALR  R12,0                  ESTABLISH BASE REGISTER
           USING *,R12
           OPEN  PRINTER,CONSOLE
           SUBSID NOTIFY,NAME=DLIID      MAKE DL/I SUBSYSTEM KNOWN
           LTR   R15,R15
           BZ    SUBOK                  NO ERRORS DETECTED: CONTINUE
           MVC   IOAREA(L'SUBEMSG),SUBEMSG
           LA    R7,L'SUBEMSG
           BAL   R3,OUTPUT              PRINT SUBSID ERROR MESSAGE
           CANCEL
  SUBOK    LA    R8,MPSXPCC
           USING IJBXPCCB,R8
           LOCK  DTL,FAIL=WAIT          SET LOCK
           XPCC  XPCCB=(R8),FUNC=IDENT   FIND OUT IF MPS IS ACTIVE
           LTR   R15,R15
           BZ    NOMPS                  MPS IS NOT ACTIVE
           TM    IJBXRETC,IJBXDAPP+IJBXAPSP
           BZ    NOMPS                  MPS IS NOT ACTIVE
           LA    R2,=C'DLZMPI00'        MPS IS ACTIVE: USE DLZMPI00
           MVC   IOAREA(L'MPSMSG),MPSMSG
           LA    R7,L'MPSMSG
           BAL   R3,OUTPUT
           B     PGMEX
  NOMPS    LA    R2,=C'DLZRRC00'        MPS NOT ACTIVE: USE DLZRRC00
           MVC   IOAREA(L'NOMPSMSG),NOMPSMSG
           LA    R7,L'NOMPSMSG
           BAL   R3,OUTPUT
  PGMEX    XPCC XPCCB=(R8),FUNC=TERMIN   EXECUTE MPS OR BATCH
           UNLOCK DTL                   REMOVE LOCK
           SUBSID REMOVE,NAME=DLIID
           FETCH (R2)                   FETCH APPROPRIATE PHASE
  OUTPUT   PUT   PRINTER                PRINT MESSAGES
           PUT   CONSOLE                DISPLAY MESSAGES
           CLOSE PRINTER,CONSOLE
           BR    R3
  MPSMSG   DC    C'MPS ACTIVE - WILL EXECUTE DLZMPI00'
  NOMPSMSG DC    C'MPS NOT ACTIVE - WILL EXECUTE DLZRRC00'
  SUBEMSG  DC    C'SUBSID NOTIFY=DLI ERROR - TERMINATE'
  DLIID    DS    0CL8                   DL/I SUBSYSTEM ID
           DC    CL7'DLI    '
           DC    X'00'
  IOAREAC  DS    0CL81
           DC    X'F1'
  IOAREA   DC    CL80' '
  MPSXPCC  XPCCB APPL=SYSDLIO1,TOAPPL=SYSDLIB1,BUFFER=(B1,B1LN)
  B1       DS    CL60
  B1LN     EQU   *-B1
  R2       EQU   2
  R3       EQU   3
  R7       EQU   7
  R8       EQU   8
  R12      EQU   12
  R15      EQU   15
  PRINTER  DTFDI DEVADDR=SYSLST,IOAREA1=IOAREAC,RECSIZE=81
  CONSOLE  DTFCN DEVADDR=SYSLOG,IOAREA1=IOAREA,BLKSIZE=80,               X
                 RECSIZE=(7),RECFORM=UNDEF,MODNAME=DLZCONSL
  DTL      DTL   NAME=DLZMPS0,CONTROL=E,LOCKOPT=1,                       X
                 KEEP=NO,OWNER=TASK
           MAPXPCCB
           END
```

Figure 20. Sample Program DLZCTRL

------------------ General-use programming interface -------------------

8.4.3 The New GSTA Call

The new GSTA call *(get statistics call)* returns to the application a block containing statistics related to system operations and to the application from which the call was issued.

Applications referencing DL/I control blocks and control areas via GSCD must operate in the same address space as DL/I. Applications using GSCD for statistics retrieval that you want to run in a VSE address space other than that of DL/I will have to be modified to use the new GSTA call function.

GSTA is valid from all DL/I environments, except for the following:

° GSTA is not supported for the HLPI interface.

° GSTA is not supported for the RPG interface.

° GSTA is not supported for remote PSBs (but GSTA *is* supported for *extended* remote PSBs).

8.4.4 GSTA Call Format

You can issue the GSTA call in assembler language, COBOL, or PL/I application programs, by using the standard DL/I call format with the following six parameters:

1. "5" as parameter count (*parmcnt*).
2. "GSTA" as the function code.
3. A valid PCB.
4. The statistics buffer.
5. The statistics buffer length (*bufflen*).
6. A return code field.

For an example of how to use the GSTA call in an assembler application program, refer to Figure 21.

```
PGMSTART     ...                Start of your application program
             ...
           MVC   GSTAPCB,0(1)   Get PCB address for GSTA call
           USING STBF,6         Establish addressability for DLZSTBF
           LA    6,BUFFER
             ...
           LA    1,GSTAPARM     Load GSTA call parameter list
           CALL  ASMTDLI        DL/I GSTA call
           L     15,GSTARCOD
           LTR   15,15          Check return code
           BNZ   GSTAERR        GSTA call not successful
             ...
GSTAERR      ...
             ...
*-----------------*
*  constant area  *
*-----------------*
             ...
GSTAPARM   DC    A(PARMCNT)      Parameter-count address
           DC    A(GSTAFUNC)     Call function address
GSTAPCB    DC    A(0)            Address of valid PCB
           DC    A(BUFFER)       Buffer start address
           DC    A(BUFFLEN)      Buffer length address
           DC    A(GSTARCOD)     Return code address
PARMCNT    DC    F'5'            Parameter count: must be 5
GSTAFUNC   DC    CL4'GSTA'       Call function: must be GSTA
BUFFER     DS    CL108           Statistics buffer
BUFFLEN    DC    F'108'          Length of statistics buffer
GSTARCOD   DC    F'0'            Four-byte area for return code
           DLZSTBF               Macro call for DLZSTBF
             ...
           END                   End of program
```

Figure 21. How to Invoke the New GSTA Call in an Assembler Program

8.4.5 GSTA Return Codes

Upon return, GSTARCOD contains the four-byte return code *rc*:

rc dec	rc (hex)	Explanation and Returned Data
0	(0000)	The GSTA call was successful.

		DL/I returns the full statistics.
4	(0004)	The passed buffer length (bufflen) is less than the statistics buffer length (108 bytes). DL/I returns the first bufflen bytes of the statistics only.
8	(0008)	The passed buffer length (bufflen) is more than the statistics buffer length (108 bytes). DL/I returns the statistics to the low-address part of the buffer.
12	(000C)	The PSB I/O work area is too small. This usually means that an ACBGEN run is required for the application PSB. No data is returned.
16	(0010)	The passed buffer length (bufflen) is not positive or is greater than 32K. No data is returned.
20	(0014)	The number of specified parameters (parmcnt) is greater than 5. No data is returned.
24	(0018)	The application runs with remote PSB. No data is returned.

Figure 22. GSTA Call Return Codes

|--------------- End of General-use programming interface ---------------|

--------------- Product-sensitive programming interface ----------------

8.4.6 Layout of the Statistics Buffer

With the new DL/I macro DLZSTBF, the returned statistics information can be referenced:

```
DLZSTBF   DSECT
STBF      DS   0F       Statistics buffer
*
STBFS     DS   0F       System statistics
STBFTSKC  DS   PL8      Total number of PSB scheduling calls
STBFDLOC  DS   PL4      Total number of program isolation
*                       deadlock occurrences
STBFCMTC  DS   PL4      Number of times at current max task
STBFPDUP  DS   PL4      Number of duplicate PSBs created
STBFRQCT  DS   F        Number of requests received
*                       by the buffer handler
STBFINPL  DS   F        Number of requests satisfied
*                       from buffer pool
```

```
|       STBFRDST  DS   F        Number of read requests issued         |
|       STBFALTR  DS   F        Number of buffer alter requests received |
|       STBFOSWT  DS   F        Number of writes issued                |
|       STBFBKWT  DS   F        Number of blocks written               |
|       STBFNWBK  DS   F        Number of new blocks created in pool   |
|       STBFCHWT  DS   F        Number of chained writes issued        |
|       STBFCHBK  DS   F        Number of blocks written on write chain |
|       STBFISTL  DS   F        Number of retrieve by key calls        |
|       STBFIGET  DS   F        Number of GN calls for KSDS received   |
|       STBFWERR  DS   X        Number of permanent write error buffers |
|       *                       in the pool                            |
|       STBFWERT  DS   X        Largest number of write error buffers  |
|       *                       ever in the pool                       |
|                 DS   H        Reserved                               |
|       STBFSEND  DS   0F       End of system statistics               |
|       *                                                              |
|       STBFJ     DS   0F       Job statistics                         |
|       STBFGU    DS   F        Number of GU   calls issued            |
|       STBFGN    DS   F        Number of GN   calls issued            |
|       STBFGNP   DS   F        Number of GNP  calls issued            |
|       STBFGHU   DS   F        Number of GHU  calls issued            |
|       STBFGHN   DS   F        Number of GHN  calls issued            |
|       STBFGHNP  DS   F        Number of GHNP calls issued            |
|       STBFISRT  DS   F        Number of ISRT calls issued            |
|       STBFDLET  DS   F        Number of DLET calls issued            |
|       STBFREPL  DS   F        Number of REPL calls issued            |
|       STBFCHKP  DS   F        Number of CHKP calls issued            |
|       STBFJEND  DS   0F       End of job related statistics          |
|       STBFEND   DS   0F       End of buffer                          |
|       *                                                              |
|       STBFLEN   EQU  STBFEND-STBF     Length of statistics buffer    |
|       STBFSLEN  EQU  STBFSEND-STBFS   Length of system statistics    |
|       STBFJLEN  EQU  STBFJEND-STBFJ   Length of job statistics       |
|                                                                      |
|                                                                      |
|______________________________________________________________________|
```

Figure 23. Layout of the Statistics Buffer DLZSTBF.
In the job statistics, the indicated numbers of DL/I calls include DL/I internal calls.

|------------ End of Product-sensitive programming interface ------------|

9.0 Chapter 9. DL/I 1.10 Support for VSE/ESA 1.3

This chapter contains General-use Programming Interface and Associated Guidance Information.

This chapter describes in detail:

- "DL/I Applications above the 16MB Line of Storage" in topic 9.1
- "VSE/VSAM HS-Buffers above the 16MB Line of Storage" in topic 9.2
- "Virtual Disk Exploitation" in topic 9.3
- "DL/I Run Statistics (DLZSTTL) Enhancements" in topic 9.4

Notes:

1. If you use the MPS Restart facility, your partition must not be larger than 16MB.

2. When using the CICS/VSE***** 2.2 distributed program link (DPL) function, the following DL/I commands are restricted in their use for application programs running in the server region:

   ```
   EXEC DLI TERM
   CALL DL/I TERM
   EXEC DLI CHKP
   CALL DL/I CHKP
   ```

   For more information on DPL, refer to the manual *CICS/VSE 2.2 Release Guide*.

3. With CICS/VSE 2.2, the name of the CICS intercommunication mirror program has been changed from DFHMIR to DFHMIRS. However, DL/I 1.10 still accepts the old name DFHMIR in the DLZACT TYPE=PROGRAM statement, but creates an entry for DFHMIRS in the Application Control Table (ACT) phase. To avoid an MNOTE in the ACT generation, specify DFHMIRS in the DLZACT TYPE=PROGRAM statement.

   Using REMOTE=YES in the DLZACT TYPE=CONFIG statement generates the equivalent of a DLZACT TYPE=PROGRAM,PGMNAME=DFHMIRS statement, which includes the PSB names of all PSBs that are in the DL/I online nucleus.

Subtopics:

- 9.1 DL/I Applications above the 16MB Line of Storage
- 9.2 VSE/VSAM HS-Buffers above the 16MB Line of Storage
- 9.3 Virtual Disk Exploitation
- 9.4 DL/I Run Statistics (DLZSTTL) Enhancements

9.1 DL/I Applications above the 16MB Line of Storage

CICS***** DL/I online applications, and DL/I batch and DL/I MPS batch applications can be executed in virtual address space that is allocated above the 16MB line of storage (also referred to as *31-bit addressing*). This is possible for all applications that are implemented in a language with 31-bit addressing capability. Languages which allow 31-bit addressing are:

- COBOL II
- High Level Assembler

Because DL/I applications can now run above the 16MB line of storage, virtual storage below the 16MB line of storage is freed. This leaves more space for DL/I data buffers and for other CICS applications.

Note that DL/I itself is limited to 24-bit addressing mode (AMODE=24). That is, all data and parameters passed to DL/I must be located below the 16MB line of storage.

If you want to use 31-bit addressing, consider the following:

- For COBOL II applications, the 31-bit addressing support is provided for programs that are coded with the HLPI or CALL interface. Refer to "Creating COBOL II Applications with 31-Bit Capability" in topic 9.1.5.
- The assembler language has no HLPI interface to DL/I. Therefore, the new support is restricted to the CALL interface.
- Assembler programs with 31-bit addressing capability should be created by using the High Level Assembler. Refer to "Creating Assembler Applications with 31-Bit Capability" in topic 9.1.6.
- CICS/VSE* macro level programs cannot run in 31-bit mode. Therefore, to exploit the extended addressability, CICS/VSE online programs must be implemented with the CICS command level interface.
- User-created exit routines (such as randomizing and compression routines) cannot run in 31-bit mode.

Subtopics:

- 9.1.1 Invocation of Applications
- 9.1.2 COBOL II and COBOL Members
- 9.1.3 High Level Assembler and Assembler Members
- 9.1.4 Migrating Applications
- 9.1.5 Creating COBOL II Applications with 31-Bit Capability
- 9.1.6 Creating Assembler Applications with 31-Bit Capability
- 9.1.7 Placing DL/I Parameter Definitions Below the 16MB Line of Storage

9.1.1 Invocation of Applications

The job control requirements to invoke a DL/I application in the CICS/VSE online, DL/I batch, or DL/I MPS environment remain unchanged.

The execution of an application in 24-bit or 31-bit address space is controlled by the new AMODE and RMODE attributes, where:

- **AMODE** defines the addressing mode in which the DL/I application will get control.
- **RMODE** defines where the application can be loaded in virtual storage.

The AMODE and RMODE attributes can be set at compile time and link-edit time by means of control statements. For detailed information on AMODE and RMODE, refer to the manual *VSE/ESA Extended Addressability*.

Figure 24 shows the possible AMODE and RMODE combinations of a load module, and the effect of the combinations on the addressing mode and application residence.

Attribute for		Addressing Mode	Residence of the Application
AMODE	RMODE		
24	24	24-bit	Below 16MB line
31	24	31-bit	Below 16MB line
ANY	24	Can be invoked in 24-bit and 31-bit addressing mode.	Below 16MB line
31	ANY	31-bit	Above 16MB line, if possible

Figure 24. Valid combinations of AMODE and RMODE

9.1.2 COBOL II and COBOL Members

The DL/I 1.10 distribution tape has new members that contain:

- COBOL II sample programs, and
- Job control to make the COBOL II sample applications executable in 31-bit address space.

The DOS/VS COBOL source that was distributed with previous DL/I releases has been adapted to VS COBOL II standards. That is, the COBOL and COBOL II members are functionally identical. For example, the:

COBOL II member DLZCB210 (HLPI load program)
executes like the
COBOL member DLZCBL10 (HLPI load program).

The following lists the equivalent COBOL II and COBOL members.

COBOL II	COBOL	Description
DLZCB2MP	DLZCBMAP	Mapping module
DLZCB210	DLZCBL10	HLPI load program
DLZCB220	DLZCBL20	HLPI batch print program
DLZCB230	DLZCBL30	Online HLPI sample application program
DLZCB240	DLZCBL40	Load program
DLZCB250	DLZCBL50	Batch print program
DLZCB260	DLZCBL60	Online sample application program

For more information on the COBOL members, refer to the manual *DL/I Guide for New Users*.

9.1.3 High Level Assembler and Assembler Members

The DL/I 1.10 distribution tape has new members that contain:

- High Level Assembler programs, and
- Job control to make the High Level Assembler applications executable in 31-bit address space.

The assembler samples that were distributed with previous DL/I releases have been adapted for 31-bit execution.

The following lists the equivalent assembler and High Level Assembler members.

High Level Assembler	Assembler	Description	Notes
DLZHMAP	DLZMAP	Mapping module	
DLZHLA40	DLZSAM40	Load program	(1) (2)
DLZHLA50	DLZSAM50	Batch print program	(1) (2)
DLZHLA60	DLZSAM60	Online sample application program	(1) (3)

Notes:

(1) The High Level Assembler requires an exit routine to allow the inclusion of VSE/ESA***** E-deck macros. The exit module EDECKXIT in the sample corresponds to the skeleton SKEDECKX provided in the ICCF library 59. However, you can also write your own exit routine. For details, refer to the manual *High Level Assembler Programmer's Guide*.

(2) The members DLZHLA40 and DLZHLA50 each produces two load modules. The function of these modules is:

1. Main phase with DL/I functions above the 16MB line of storage.

2. Phase to perform I/O operations below the 16MB line of storage. This phase is called by the main phase.

(3) Because *31-bit addressing* is not supported at the CICS macro level, the assembler member DLZSAM60 has been converted to DLZHLA60 using CICS command level.

For more information on the assembler members, refer to the manual *DL/I Guide for New Users*.

9.1.4 Migrating Applications

The DL/I support for 31-bit addressing does not affect any of your existing DL/I applications. That is, existing applications continue to run in 24-bit addressing mode and 24-bit address space. This is independent of the partition size.

On the other hand, if you want to *enable* existing DL/I applications for 31-bit addressing, follow the instruction under:

- "Enabling COBOL Applications" in topic 9.1.4.1
- "Enabling Assembler Applications" in topic 9.1.4.2

Subtopics:

- 9.1.4.1 Enabling COBOL Applications
- 9.1.4.2 Enabling Assembler Applications

9.1.4.1 Enabling COBOL Applications

To enable a DL/I application that is implemented in DOS VS COBOL language, proceed as follows:

1. Convert the program to VS COBOL II standard.

For information, refer to the manual *VS COBOL II Migration Guide for VSE*.

2. Re-compile and re-link the source code according to the procedures given in "Creating COBOL II Applications with 31-Bit Capability" in topic 9.1.5.

9.1.4.2 Enabling Assembler Applications

To enable a DL/I application that is implemented in assembler language, proceed as follows:

1. Verify that the source code generally adheres to the programming standards required for 31-bit addressing. For details, refer to the manual *VSE/ESA Extended Addressability*.

2. Keep the DL/I data and parameters below the 16MB line of storage. For more information, see "Placing DL/I Parameter Definitions Below the 16MB Line of Storage" in topic 9.1.7.

3. Re-compile and re-link the source code according to the procedure given in "Creating Assembler Applications with 31-Bit Capability" in topic 9.1.6.

9.1.5 Creating COBOL II Applications with 31-Bit Capability

For execution in 31-bit address space, you can:

enable existing DL/I applications, and
create new DL/I applications

by using the following procedures and examples.

To enable a DL/I application written in COBOL II language, you have to:

1. Compile the application with the options **RENT** and **RES**.

2. Keep the DL/I data and parameters below the 16MB line of storage.

To do so, specify the compile option **DATA(24)**.

For information on how to write a COBOL II source program, refer to the manual:

° *VS COBOL II Application Programming Guide for VSE*
° *VS COBOL II Application Programming Language Reference*

The following examples (Figure 25 in topic 9.1.5.1, Figure 26 in topic 9.1.5.2, and Figure 27 in topic 9.1.5.3) illustrate the job control statements needed to enable existing, or to create new COBOL II applications for execution in 31-bit address space.

Subtopics:

- 9.1.5.1 Example: COBOL II - Online with HLPI or CALL Interface
- 9.1.5.2 Example: COBOL II - Batch and MPS Batch with HLPI
- 9.1.5.3 Example: COBOL II - Batch and MPS Batch with CALL Interface

---

9.1.5.1 Example: COBOL II - Online with HLPI or CALL Interface

```
 ________________________________________________________________________
|                                                                        |
|                                                                        |
|      // JOB CB2SAMPL                                                   |
|      // DLBL IJSYSPH,'COBOL II TRANSLATION',yy/ddd                     |
|      // EXTENT SYSPCH,extent-information                               |
ASSGN SYSPCH,DISK,VOL=volid,SHR
// LIBDEF *,SEARCH=search-library
// LIBDEF PHASE,CATALOG=catalog-library
// EXEC DFHECP1$,SIZE=...
CBL XOPTS(CICS,DLI,COBOL2),LIB,RENT,RES,DATA(24)   See Note 1.
       .
       .
  SOURCE DECK
       .
       .
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'COBOL II TRANSLATION',yy/ddd
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// OPTION NODECK,CATAL
PHASE CB2SAMPL,*
INCLUDE DFHECI
// EXEC IGYCRCTL,SIZE=...
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&

Note 1: If you use the DL/I CALL interface,
omit the DL/I parameter in the CBL XOPTS statement.
```

Figure 25. Example: COBOL II - Online with HLPI or Call Interface

9.1.5.2 Example: COBOL II - Batch and MPS Batch with HLPI

```
      // JOB CB2SAMPL
      // DLBL IJSYSPH,'COBOL II TRANSLATION',yy/ddd
      // EXTENT SYSPCH,extent-information
      ASSGN SYSPCH,DISK,VOL=volid,SHR
      // LIBDEF *,SEARCH=search-library
      // LIBDEF PHASE,CATALOG=catalog-library
      // EXEC DFHECP1$,SIZE=...
       CBL XOPTS(DLI,COBOL2),LIB,RENT,RES,DATA(24)
              .
              .
         SOURCE DECK
              .
              .
      /*
      CLOSE SYSPCH,punch
      // DLBL IJSYSIN,'COBOL II TRANSLATION',yy/ddd
      // EXTENT SYSIPT
      ASSGN SYSIPT,DISK,VOL=volid,SHR
      // OPTION NODECK,CATAL
         PHASE CB2SAMPL,*
         INCLUDE DLZBPJRA
         INCLUDE DLZLICBL
      // EXEC IGYCRCTL,SIZE=...
         ENTRY CBLCALLA
      // EXEC LNKEDT
```

```
       /&
       // JOB RESET
       CLOSE SYSIPT,reader
       /&
```

Figure 26. Example: COBOL II - Batch and MPS Batch with HLPI

9.1.5.3 Example: COBOL II - Batch and MPS Batch with CALL Interface

```
       // JOB CB2SAMPL
       // LIBDEF *,SEARCH=search-library
       // LIBDEF PHASE,CATALOG=catalog-library
       // OPTION NODECK,CATAL
          PHASE CB2SAMPL,*
          INCLUDE DLZBPJRA
       // EXEC IGYCRCTL,SIZE=...
        CBL LIB,RENT,RES,DATA(24)
                .
                .
          SOURCE DECK
                .
                .
       /*
          ENTRY CBLCALLA
       // EXEC LNKEDT
       /&
```

Figure 27. Example: COBOL II - Batch and MPS Batch with CALL

9.1.6 Creating Assembler Applications with 31-Bit Capability

For execution in 31-bit address space, you can:

enable existing DL/I assembler applications, and
create new DL/I assembler applications

by using the following procedure and examples.

To enable a DL/I application written in assembler language, you have to:

1. Compile the application by using the *High Level Assembler*.

The source program must contain the instructions:

```
AMODE 31
```

and
RMODE ANY

Note: The correct AMODE/RMODE settings alone do not guarantee that a program is suited for 31-bit execution. It is the programmer's responsibility to ensure that the assembler source itself satisfies the requirements of 31-bit addressing.

2. Keep the DL/I data and parameters below the 16MB line of storage.

For details, see "Placing DL/I Parameter Definitions Below the 16MB Line of Storage" in topic 9.1.7.

For information on using the High Level Assembler and programming techniques in the 31-bit environment, refer to the manuals:

- *High Level Assembler Programmer's Guide* and
- *VSE/ESA Extended Addressability*

The following examples illustrate the job control statements needed to enable existing, or to create new assembler applications for execution in 31-bit address space:

- Figure 28 in topic 9.1.6.1 applies to DL/I online applications.
- Figure 29 in topic 9.1.6.2 applies to DL/I batch and MPS batch applications.

Note: With regard to the LIBEXIT parameter (shown in Figure 28 in topic 9.1.6.1 and Figure 29), the High Level Assembler requires an exit routine to allow the inclusion of VSE/ESA E-deck macros. The exit module EDECKXIT in the sample corresponds to the skeleton SKEDECKX provided in the ICCF library 59. However, you can also write your own exit routine. For details, refer to the manual *High Level Assembler Programmer's Guide*.

Subtopics:

- 9.1.6.1 Example: High Level Assembler - Online
- 9.1.6.2 Example: High Level Assembler - Batch and MPS Batch

9.1.6.1 Example: High Level Assembler - Online

```
 ____________________________________________________________________
|                                                                    |
|                                                                    |
|     // JOB HLASAMPL                                                |
|     // DLBL   IJSYSPH,'ASSEMBLER TRANSLATION',yy/ddd                |
|     // EXTENT SYSPCH,extent-information                            |
|     ASSGN SYSPCH,DISK,VOL=volid,SHR                                |
|     // LIBDEF *,SEARCH=search-library                              |
|     // LIBDEF PHASE,CATALOG=catalog-library                        |
|     // EXEC DFHEAP1$,SIZE=...                                      |
|     *ASM XOPTS(CICS)                                               |
|      HLASAMPL CSECT                                                  |
|      HLASAMPL AMODE 31                                               |
|      HLASAMPL RMODE ANY                                              |
|              .                                                       |
|              .                                                       |
|         SOURCE DECK                                                  |
|              .                                                       |
|              .                                                       |
|      /*                                                              |
|      CLOSE SYSPCH,punch                                              |
|      // DLBL   IJSYSIN,'ASSEMBLER TRANSLATION',yy/ddd                |
|      // EXTENT SYSIPT                                                |
|      ASSGN SYSIPT,DISK,VOL=volid,SHR                                 |
|      // OPTION  SYM,ERRS,NODECK,CATAL                                |
|         PHASE   HLASAMPL,*                                           |
|         INCLUDE DFHEAI                                               |
|      // EXEC ASMA90,SIZE=(...),PARM='EXIT(LIBEXIT(edeckxit))'        |
|      // EXEC LNKEDT                                                  |
|      /&                                                              |
|      // JOB RESET                                                    |
|      CLOSE SYSIPT,reader                                             |
|      /&                                                              |
|                                                                      |
|                                                                      |
|______________________________________________________________________|
```

Figure 28. Example: High Level Assembler - Online

9.1.6.2 Example: High Level Assembler - Batch and MPS Batch

```
 ______________________________________________________________________
|                                                                      |
|                                                                      |
|      // JOB HLASAMPL                                                 |
|      // LIBDEF *,SEARCH=search-library                               |
|      // LIBDEF PHASE,CATALOG=catalog-library                         |
|      // OPTION CATAL,NODECK                                          |
|         PHASE HLASAMPL,*                                             |
|      // EXEC ASMA90,SIZE=(...),PARM='EXIT(LIBEXIT(edeckxit))'        |
|      HLASAMPL CSECT                                                  |
|      HLASAMPL AMODE 31                                               |
|      HLASAMPL RMODE ANY                                              |
|              .                                                       |
|              .                                                       |
|         SOURCE DECK                                                  |
|              .                                                       |
|              .                                                       |
|      /*                                                              |
|      // EXEC LNKEDT                                                  |
|      /&                                                              |
|                                                                      |
|                                                                      |
|______________________________________________________________________|
```

Figure 29. Example: High Level Assembler - Batch and MPS Batch

9.1.7 Placing DL/I Parameter Definitions Below the 16MB Line of Storage

Your actions depend on whether the assembler application is a:

- CICS/VSE* DL/I online application, or
- DL/I batch application

Subtopics:

- 9.1.7.1 CICS/VSE DL/I Online Application
- 9.1.7.2 DL/I Batch Application

9.1.7.1 CICS/VSE DL/I Online Application

If it is a CICS/VSE DL/I online application, no extra effort is required.

As stated above, only applications using the CICS command level interface are supported in 31-bit address space. This generally implies that all user variables and parameter lists are defined through DFHEISTG. CICS working storage defined through DFHEISTG will still be allocated below the 16MB line of storage. If the DL/I parameters are defined outside of DFHEISTG, they must be moved below the 16MB line of storage as described under "DL/I Batch Application" in topic 9.1.7.2.

9.1.7.2 DL/I Batch Application

If it is a DL/I batch application, all DL/I data and parameters must be held in storage that is dynamically obtained below the 16MB line of storage. This means that you may have to restructure the existing program where the DL/I parameters are defined.

Figure 30 is an example of how to change an existing assembler program so that the DL/I parameter definitions will be located below the 16MB line of storage, even when the program itself is executed above the 16MB line of storage.

Figure 30 basically corresponds to "Figure 1-12" given in the manual *DL/I Application Programming: CALL and RQDLI Interfaces*. There, you find further notes and explanations to the statements.

```
 ______________________________________________________________________
|        Existing Program                    Changed Program           |
|                                                                      |
|PGMSTART CSECT                      PGMSTART CSECT                    |
|                                    PGMSTART AMODE 31                 |
|                                    PGMSTART RMODE ANY                |
|         USING *,R12                         USING *,R12              |
|         SAVE  (R14,R12)                     SAVE  (R14,R12)          |
|         LR    R12,R15                       LR    R12,R15            |
|         ST    R13,SAVEAREA+4                ST    R13,SAVEAREA+4     |
|         LA    R13,SAVEAREA                  LA    R13,SAVEAREA       |
|         .                                   .                        |
|         MVC DBPCBMST(8),0,(R1)              MVC DBPCBMST(8),0(R1)    |
|         .                                   .                        |
|                                                       LA    R0,PARMEND-PARMSTRT|
|                                                       GETVIS LOC=BELOW ...     |
|                                                       LR    R5,R1              |
|                                                       USING PARMSTRT,R5        |
|                                                       MVC   PARMCT,=F'4'       |
|                                                       MVC   SSANAME,RTNAME     |
|                                                       LA    R1,PARMCT          |
|                                                       ST    R1,PARMLIST        |
|                                                       LA    R1,DLIFUNC         |
|                                                       ST    R1,FUNC            |
|         .                                             .                        |
|         MVC   DLIFUNC,GU                              MVC   DLIFUNC,GU         |
|         MVC   PCB,DBPCBDET                            MVC   PCB,DBPCBDET       |
|         LA    R1,DETSEGIO                             LA    R1,DETSEGIO        |
|         ST    R1,IOAREA                               ST    R1,IOAREA          |
|         LA    R1,SSANAME                              LA    R1,SSANAME         |
|         ST    R1,SSA                                  ST    R1,SSA             |
|         LA    R1,PARMLIST                             LA    R1,PARMLIST        |
|         CALL  ASMTDLI                                 CALL  ASMTDLI            |
|         .                                             .                        |
|         MVC   DLIFUNC,GHU                             MVC   DLIFUNC,GHU        |
|         MVC   PCB,DBPCBMST                            MVC   PCB,DBPCBMST       |
|         LA    R1,MSTSEGIO                             LA    R1,MSTSEGIO        |
|         ST    R1,IOAREA                               ST    R1,IOAREA          |
|         MVI   SSANAME+8,C' '                          MVI   SSANAME+8,C' '     |
|         LA    R1,PARMLIST                             LA    R1,PARMLIST        |
|         CALL  ASMTDLI                                 CALL  ASMTDLI            |
|         .                                             .                        |
|         MVC   DLIFUNC,GHN                             MVC   DLIFUNC,GHN        |
|         MVI   PARMCT+3,3                              MVI   PARMCT+3,3         |
|         LA    R1,PARMLIST                             LA    R1,PARMLIST        |
|         CALL  ASMTDLI                                 CALL  ASMTDLI            |
|         .                                             .                        |
|________________________________________________________________________________|
 ________________________________________________________________________________
|         MVC   DLIFUNC,REPL                            MVC   DLIFUNC,REPL       |
|         LA    R1,PARMLIST                             LA    R1,PARMLIST        |
|         CALL  ASMTDLI                                 CALL  ASMTDLI            |
|         .                                             .                        |
|         L     R13,4(R13)                              L     R13,4(R13)         |
|         RETURN (R14,R12)                              RETURN (R14,R12)         |
|*    CONSTANT AREA                           *    CONSTANT AREA                |
|                                                                                |
|PARMLIST DC    A(PARMCT)                                                       |
|FUNC     DC    A(DLIFUNC)                                                      |
|PCB      DC    A(0)                                                            |
|IOAREA   DC    A(0)                                                            |
|SSA      DC    A(0)                                                            |
|         .                                                                      |
|DBPCBMST DC    F'0'                          DBPCBMST DC    F'0'               |
|DBPCBDET DC    F'0'                          DBPCBDET DC    F'0'               |
|        .                                            .                         |
|PARMCT   DC    F'4'                                                            |
|DLIFUNC  DC    CL4'    '                                                       |
|GU       DC    CL4'GU  '                     GU       DC    CL4'GU  '          |
|GHU      DC    CL4'GHU '                     GHU      DC    CL4'GHU '          |
|GHN      DC    CL4'GHN '                     GHN      DC    CL4'GHN '          |
|REPL     DC    CL4'REPL'                     REPL     DC    CL4'REPL'          |
|CHKP     DC    CL4'CHKP'                     CHKP     DC    CL4'CHKP'          |
|         .                                             .                        |
|SSANAME  DS    0CL26                         RTNAME   DS    0CL26              |
|ROOT     DC    CL8'ROOT    '                 ROOT     DC    CL8'ROOT    '      |
|         DC    CL1'('                                 DC    CL1'('             |
|         DC    CL8'KEY     '                          DC    CL8'KEY     '      |
|         DC    CL2' ='                                DC    CL2' ='            |
|NAME     DC    CL6'vvvvvv'                   NAME     DC    CL6'vvvvvv'        |
|         DC    CL1')'                            DC    CL1')'            |
|         .                                       .                       |
|MSTSEGIO DS    CL100                                                     |
|DETSEGIO DS    CL100                                                     |
|SAVEAREA DC    18F'0'                   SAVEAREA DC    18F'0'            |
|         .                                                               |
|                                        * DEFINITIONS FOR DL/I PARM. LIST |
|                                                                         |
|                                                 DSECT                   |
|                                        PARMSTRT DS    0H                |
|                                        PARMLIST DS    A(0)              |
|                                        FUNC     DS    A(0)              |
|                                        PCB      DS    A(0)              |
|                                        IOAREA   DS    A(0)              |
|                                        SSA      DS    A(0)              |
|_________________________________________________________________________|
 _________________________________________________________________________
|                                        PARMCT   DS    F                 |
|                                        DLIFUNC  DS    CL4               |
|                                        SSANAME  DS    CL26              |
|                                        MSTSEGIO DS    CL100             |
|                                        DETSEGIO DS    CL100             |
|                                        PARMEND  DS    0H                |
|                                                 .                       |
|PCBNAME  DSECT                          PCBNAME  DSECT                   |
|DBPCBDBD DS    CL8                      DBPCBDBD DS    CL8               |
|DBPCBLEV DS    CL2                      DBPCBLEV DS    CL2               |
|         .                                       .                       |
|         .                                       .                       |
|         .                                       .                       |
|         END                                     END                     |
|_________________________________________________________________________|
```

Figure 30. Re-allocation of DL/I Parameters in an Assembler Program

9.2 VSE/VSAM HS-Buffers above the 16MB Line of Storage

DL/I exploits the VSE/VSAM support that allows you to allocate input/output buffers above the 16MB line of storage. Specifically, VSE/VSAM input/output buffers may be allocated above the 16MB line of storage for the use of DL/I:

- KSDS index files, and
- HISAM KSDS and ESDS data files, and
- SHISAM KSDS data files.

In the following, these buffers are referred to as *HS-buffers*.

Up to and including DL/I 1.9, you were able to control the allocation of HS-buffers by specifying the *number* of buffers in the *HSBFR* parameter, or by using a default. With the new support in DL/I 1.10, you can also specify the **residence** of these buffers.

Depending on the number of HS-buffers defined above the 16MB line of storage, the VSE/VSAM input/output operations are reduced for the above listed files. The CPU time overhead from moving data due to the change from VSE/VSAM LOCATE mode to VSE/VSAM MOVE mode is negligible.

Subtopics:

- 9.2.1 Considerations
- 9.2.2 Compatibility of Existing Applications
- 9.2.3 Invocation

9.2.1 Considerations

- The buffer pool management for HDAM/HIDAM ESDS files (VSAM user buffering) is not changed. That is, buffer allocation beyond 16MB is not allowed.

- The new parameters HSMODE=ANY|BELOW are also reflected in the DL/I interactive functions, IMF and IUG.

  For information on IMF and IUG, refer to "DL/I IMF and IUG Functions" in topic 5.3, and the manuals:

  - *DL/I Guide for New Users*
  - *DL/I Interactive Resource Definition and Utilities*

9.2.2 Compatibility of Existing Applications

The DL/I support for allocating HS-buffers above the 16MB line of storage does not affect your existing DL/I applications. That is, if the parameter HSMODE is *not* specified (this will be the case in all existing DL/I batch jobs and online ACT generations), DL/I assumes the default HSMODE=BELOW. This places the HS-buffers below the 16MB line of storage (as was the case prior to DL/I 1.10).

On the other hand, if you want the HS-buffers to reside above the 16MB line of storage, you have to specify buffer residence as described under "Invocation" in topic 9.2.3.

9.2.3 Invocation

To specify buffer residence, use the new parameter **HSMODE** for the:

- **DL/I batch environment** in the DL/I control statement.

  The HSMODE parameter can be specified whenever the HSBFR parameter can be specified.

  For example, to allocate 50 KSDS index and data buffers in 31-bit address space for a batch program, a new DL/I batch control statement could be:

```
DLI,SAMPLPRG,PSB1,,HDBFR=(10),HSBFR=(50,50,,INDEX),HSMODE=ANY
```

° **DL/I online environment** during online nucleus generation in the DLZACT TYPE=CONFIG macro statement.

For example, to allocate 50 KSDS index and data buffers in 31-bit address space for an online program, you have to provide *two statements* as follows:

1. A DLZACT TYPE=CONFIG statement to specify the *residence* of the HS-buffers in 31-bit address space:

```
DLZACT TYPE=CONFIG,
     REMOTE=NO,
     BFRPOOL=3,
     HSMODE=ANY,
     MAXTASK=10,
     CMAXTSK=5,
     SLC=DLZSLC01,
     PI=YES
```

2. A DLZACT TYPE=BUFFER statement to specify the *number* of the HS-buffers:

```
DLZACT TYPE=BUFFER,
     HDBFR=(20,HIDAM2),
     HSBFR=(50,50,,INDEX2)
```

An equivalent statement is required for each individual data base.

Applicable for both environments:

HSMODE=ANY
first tries to allocate buffers above the 16MB line of storage. If the attempt fails (all storage above the 16MB line of storage consumed or partition too small), HSMODE=ANY tries to allocate storage below the 16MB line of storage.

HSMODE=BELOW (the default)
specifies that buffers are to be located below the 16MB line of storage.

9.3 Virtual Disk Exploitation

All DL/I utilities can now allocate their work files on a virtual disk (that is, in virtual data space). The utility jobs do not have to be changed, however the job control statements (ASSGN, DLBL and EXTENT) must address the virtual disk instead of the real device. Allocating work files on a virtual disk significantly improves performance by reducing input/output operations to a real device.

The following DL/I utilities can profit from virtual disk utilization:

- Initial Load
- Partial Load
- DLZURGL0 HD Reorganization Reload
- DLZURPR0 Data Base Pre-reorganization
- DLZURGS0 Data Base Scan
- DLZURG10 Data Base Prefix Resolution
- DLZURGP0 Data Base Prefix Update
- DLZPRCTn Partial Data Base Reorganization
- DLZUCUM0 Data Base Change Accumulation

Because virtual disks are *not* permanent, they should only be used for files that can be recovered in case of loss (for example, after IPL).

For more information on virtual disk support, refer to the manual *VSE/ESA Extended Addressability*.

9.4 DL/I Run Statistics (DLZSTTL) Enhancements

The program DLZSTTL has been enhanced to provide new and improved information about the data base activities. This information can assist the data base administrator to optimize the DL/I data base allocation.

The changes are as follows:

- New summary of DL/I calls. Calls are accumulated from DL/I startup to the execution of DLZSTTL.

- Updated buffer pool call summary.

- Important information from the buffer pool statistics are shown per subpool.

- New statistics per data base. VSE/VSAM information is shown for non-HD data bases.

Figure 31 shows an example of the output of the DL/I Run Statistics program DLZSTTL.

```
DLZSTTL =================================================================
DLZSTTL     DL/I  DATA BASE- AND SUBPOOL-STATISTICS      92/094    9:23:33
DLZSTTL =================================================================
DLZSTTL     DL/I RUN STATISTICS
DLZSTTL --------------------------------------------------------
DLZSTTL - NUMBER OF PSB SCHEDULING CALLS                     6
DLZSTTL - NUMBER OF TIMES AT PI DEADLOCK                     -
DLZSTTL - NUMBER OF TIMES IN PI WAIT                         4
DLZSTTL - NUMBER OF TIMES AT CURRENT MAX TASK                 -
DLZSTTL - NUMBER OF DUPLICATE PSBS CREATED                    -
DLZSTTL --------------------------------------------------------
DLZSTTL - NUMBER OF GU   CALLS ISSUED                         -
DLZSTTL - NUMBER OF GN   CALLS ISSUED                         1
DLZSTTL - NUMBER OF GNP  CALLS ISSUED                         -
DLZSTTL - NUMBER OF GHU  CALLS ISSUED                         2
DLZSTTL - NUMBER OF GHN  CALLS ISSUED                       120
DLZSTTL - NUMBER OF GHNP CALLS ISSUED                         -
DLZSTTL - NUMBER OF ISRT CALLS ISSUED                        11
DLZSTTL - NUMBER OF DLET CALLS ISSUED                         4
DLZSTTL - NUMBER OF REPL CALLS ISSUED                         3
DLZSTTL - NUMBER OF CHKP CALLS ISSUED                         -
DLZSTTL --------------------------------------------------------
DLZSTTL     DL/I BUFFER POOL STATISTICS
DLZSTTL --------------------------------------------------------
DLZSTTL - NUMBER OF BUFFER HANDLER REQUESTS                1221
DLZSTTL - NUMBER OF REQUESTS SATISFIED FROM POOL            770
DLZSTTL - NUMBER OF BUFFER READ REQUESTS ISSUED             309
DLZSTTL - NUMBER OF BUFFER ALTER REQUESTS ISSUED             21
DLZSTTL - NUMBER OF BUFFER WRITE REQUESTS ISSUED             11
DLZSTTL - NUMBER OF NEW BLOCKS CREATED IN POOL                -
DLZSTTL - NUMBER OF CHAINED WRITE REQUESTS ISSUED             -
DLZSTTL - NUMBER OF BLOCKS WRITTEN ON WRITE CHAIN             -
DLZSTTL - NUMBER OF RETRIEVALS BY KEYED CALLS                81
DLZSTTL - NUMBER OF RETRIEVALS BY GETNEXT CALLS              46
DLZSTTL - NUMBER OF RETRIEVALS BY RBA CALLS                   -
DLZSTTL - NUMBER OF PERMANENT WRITE ERRORS                    -
DLZSTTL - NUMBER OF BUFFER PURGE CALLS                        5
DLZSTTL =================================================================
DLZSTTL     DL/I COMMON SUBPOOL AND DATA BASE INFORMATION
DLZSTTL =================================================================
DLZSTTL - NUMBER OF SUBPOOLS                                  4
DLZSTTL
DLZSTTL -  DL/I RELATED INFORMATION PER SUBPOOL -
DLZSTTL
DLZSTTL SUB-     NUMBER  NUMBER OF    BUFFER      NUMBER OF  REQ. SATISFD     NUMBE
DLZSTTL POOL    OF DMBS    BUFFERS      SIZE    BH REQUESTS     FROM POOL  READ REQ
DLZSTTL ----I----------I----------I---------I--------------I-------------I---------
DLZSTTL    1          1          8       512              -             -
DLZSTTL    2          1         10      2048           1064           738
DLZSTTL    3          2         20      2048             39            32
DLZSTTL    4          2         32      4096              -             -
DLZSTTL
DLZSTTL =================================================================
DLZSTTL
DLZSTTL - INFORMATION PER DATA BASE -
DLZSTTL
DLZSTTL ----------(DL/I)---------I I------(VSAM)------I-----------(VSAM FOR HISAM,S
DLZSTTL
DLZSTTL DBD-    DMB-   SUB-  BUF- W   CI-    NUMBER OF   NUMBER OF   NUMBER OF    NU
DLZSTTL NAME    ORGAN. POOL  SIZE    SIZE        EXCPS   CI-SPLITS   CA-SPLITS    R
DLZSTTL -------I------I----I-----I-I-----I------------I-----------I-----------I----
DLZSTTL D08416A HDAM      1   512     512            -
DLZSTTL DD05264 HDAM      3  2048         (ACB CLOSED)
DLZSTTL KAHIDI  INDEX     4  4096         (ACB CLOSED)
DLZSTTL KAHIDAM HIDAM     4  4096         (ACB CLOSED)
DLZSTTL STDIX1P INDEX               2048            -           -           -
DLZSTTL STDIDBP HDAM      3  2048    2048            6
DLZSTTL STDCDBP HDAM      2  2048    2048          314
DLZSTTL STDCX2P INDEX               2048            2           -           -
DLZSTTL STDCX1P INDEX               2048           10           -           -
DLZSTTL -------I------I----I-----I-I-----I------------I-----------I-----------I----
DLZSTTL SUBPOOL           2                        314
DLZSTTL SUBPOOL           3                          6
DLZSTTL -------I------I----I-----I-I-----I------------I-----------I-----------I----
DLZSTTL TOTAL                                      332           -           -
DLZSTTL -------I------I----I-----I-I-----I------------I-----------I-----------I----
DLZSTTL
DLZSTTL   N O T E
DLZSTTL      CI-SIZE         = CI-SIZE OF THE DATA BASE
DLZSTTL      BUF-SIZE        = BUFFER-SIZE OF THE SUBPOOL
DLZSTTL      W (WARNING)     = *, IN CASE CI-SIZE NOT EQUAL BUFFER-SIZE
DLZSTTL
DLZSTTL   R E M A R K S
DLZSTTL      THE VSAM STATISTICS VALUES ARE FROM THE LAST START OF THE DATA BASE.
DLZSTTL
DLZSTTL =================================================================
DLZSTTL     DL/I STATISTICS REPORT COMPLETE
DLZSTTL =================================================================
```

Figure 31. Example: Output of the DL/I Run Statistics Program (DLZSTTL)

If in this statistics an index component is assigned to a subpool, it means that this index database has an insert processing option. It does *not* mean that the index buffers are allocated to that subpool. Allocation of index buffers should be specified through:

```
TYPE=BUFFER,HSBFR=...
```

10.0 Chapter 10. DL/I 1.9 Support for VSE/ESA 1.1

Subtopics:

- 10.1 Dynamic Partitions
- 10.2 Database I/O Error Handling
- 10.3 DL/I in a CICS/XRF Environment

10.1 Dynamic Partitions

DL/I runs in static partitions as before. DL/I batch and MPS jobs can also run in VSE/ESA***** dynamic partitions, but all restrictions for dynamic partitions apply. For DL/I, this means:

° MPS restart is not possible, because Checkpoint/Restart is not supported in a dynamic partition.

° The following utilities may run in dynamic partitions:

  - ACBGEN (DLZUACB0), and

- Data Base Pre-Reorganization (DLZURPR0),

but the system logical units SYSRDR, SYSIPT, SYSPCH, and SYSLST may not be assigned to a disk.

For example, you may *not* run a job in a dynamic partition if you

assign SYSPCH to a disk in the first job step, and then reassign the disk to SYSIN in a following job step to process the output of the first job step.

For a detailed description of VSE/ESA dynamic partitions and applicable restrictions, refer to the manual *VSE/ESA Planning*.

10.2 Database I/O Error Handling

The following description includes the changes made by APAR PN09116.

DL/I provides an enhanced handling of damaged DL/I databases, providing CICS***** journaling is active.

If an I/O error occurs, DL/I issues the message DLZ004I or DLZ005I, and depending on the DLIOER parameter, DL/I either:

° Shuts down the DL/I CICS system and saves the database stop record, or
° Stops the database, saves a database stop record, and continues operation.

After warm start, emergency restart, or takeover, the database stop record is used to notify the user that the database has been damaged.

CICS/VSE***** Version 2 provides new SIT parameters:

```
XRF=YES|NO
DLIOER=ABEND|CONTINUE
```

For a detailed description of the CICS XRF parameters, refer to the manual *CICS/VSE 2.1 XRF Guide*.
The combination of the XRF and DLIOER parameter determines the action taken by the DL/I CICS system after an I/O error occurs with a DL/I database:

DLIOER=ABEND and XRF=NO After the message DFH4540, the DL/I CICS system will be terminated.

DLIOER=ABEND and XRF=YES After the message DFH4540, the active DL/I CICS system will be terminated. Messages are issued by the alternate DL/I CICS system (if any) to inform the operator how to continue.

DLIOER=CONTINUE and XRF=NO|YES After the message DFH4541, the active DL/I CICS system continues processing and the damaged database will be stopped. After a CICS warm start, emergency restart or takeover, the database cannot be accessed until it is repaired. Messages are issued during DL/I initialization to inform the operator how to continue. See also Chapter 11, "New Messages" in topic 11.0.

For each damaged data set of a data base stopped due to an I/O error, DL/I writes a database stop record into the CICS system journal. The database stop record is saved into the CICS restart data set. As long as the database remains stopped, any PSB that references this database cannot be scheduled. Your actions:

° Use the DL/I STOP command to close associated ACB(s) and use off-line DL/I utilities or take appropriate recovery actions to recover the database.

° To use the database again, restart the database through the DL/I STRT command.

After warm start, emergency restart, or takeover, and if you have not repaired and restarted the damaged database, DL/I issues messages DLZ141I and DLZ142A for each database stop record found in the CICS restart data set.
Message DLZ142A prompts the user to perform one of the following actions:

CONTINUE The database has been repaired and DL/I initialization continues. Automatic backout will be performed for any "inflight" log records found on the CICS system log. This (in addition to the previous executed batch forward recovery) ensures database integrity.

The database will be put into "initial" state. If OPEN=DEFERRED is coded in File Control Table (FCT), the database must be opened using the DL/I system call STRT before the database can be used again.

IGNORE The database has not been repaired and DL/I initialization continues. Because the database remains stopped, automatic backout cannot be performed during CICS start or restart for those databases that are:

° Associated with PSBs that reference the stopped database.
° Notified in the CICS message(s) DFH5723.

CANCEL DL/I initialization and CICS startup terminates.

During the open operation of the repaired database, a database *open repaired record* is written into the CICS system log and CICS removes the corresponding database *stop record* from the CICS restart data set.

Note: During a cold start, all database stop records are lost and DL/I assumes that all damaged databases have already been repaired.

Subtopics:

- 10.2.1 Repair of an I/O Error Data Base

10.2.1 Repair of an I/O Error Data Base

- If **DLIOER=ABEND** was specified in SIT, and if the DL/I CICS system abended because of an I/O error, proceed as follows:

  1. After the completion of the batch DL/I forward recovery, perform a CICS emergency restart.

  2. During the restart (after you replied with CONTINUE to the messages DLZ141I and DLZ142A), the records of tasks that were "inflight" at the time of failure will be backed out (up to and including the last complete logical unit of work - LUW).

- If **DLIOER=CONTINUE** was specified in SIT, and if the DL/I CICS system continues processing after an I/O error, the following conditions can exist:

  - Depending on the type of I/O error and the moment at which the failure occurs, DL/I returns a status code to the application program in order to abnormally terminate the task.

    If the task terminates abnormally, dynamic transaction backout (if specified) will be performed.

  - It also can happen, however, that DL/I is unable to return a status code to the application program, because the application program has already advanced. This is because the VSAM "physical write" occurs much later than the DL/I update request.

  Note that in either of the conditions, the tasks are not considered "inflight," because their task termination records exist on the system log.

  To repair the database:

  1. Before running batch DL/I forward recovery, remove the log records that relate to these tasks (those tasks that are not considered "inflight") from the system log.

     The other method is to run the DL/I backout utility after forward recovery.

  2. Restart CICS. To the messages DLZ141I and DLZ142A, reply with CONTINUE. Remember that no backout is performed.

In **general**: If you start or restart CICS without a previously performed forward recovery, reply with IGNORE to the messages DLZ141I and DLZ142A.

It is the user's responsibility to perform subsequent recovery procedures.

10.3 DL/I in a CICS/XRF Environment

VSE/ESA* provides the CICS/VSE* 2.1 Extended Recovery Facility (CICS XRF). The facility guarantees high availability by an automatic system recovery after a failure. For a detailed description of CICS XRF refer to the manual *CICS/VSE 2.1 XRF Guide*.

Within a DL/I CICS XRF environment, there is an *active* and an *alternate* DL/I CICS system:

- The active DL/I CICS system performs the normal DL/I processing.

- The alternate CICS waits in a standby mode until the active CICS terminates due to an error situation or a console command, and then takes over the DL/I CICS system including the DL/I databases.

The DL/I system of the alternate CICS is partially initialized because the DL/I databases can only be owned either by the active CICS or the alternate CICS. During takeover, the alternate CICS completes the DL/I initialization, opens the DL/I databases and performs a backout if required. If in the active DL/I CICS system an I/O error occurs with a DL/I database, the user is asked during takeover whether the DL/I initialization is to be continued or canceled. For actions, refer to"Repair of an I/O Error Data Base" in topic 10.2.1.

Subtopics:

- 10.3.1 Requirements
- 10.3.2 Setup
- 10.3.3 System Integrity

10.3.1 Requirements

- Shared Databases

  For DL/I data sets, specify DISP=SHR. For more information about sharing data sets, see the manual *CICS/VSE 2.1 XRF Guide*.

- Shared CICS System Log

  A DL/I CICS XRF environment requires CICS system journals shared by the active CICS and the alternate CICS. The CICS system journal must be defined with DISP=SHR in JCL. For details, refer to the manual

CICS/VSE 2.1 XRF Guide.

If DL/I CICS XRF system runs without a CICS system journal, DL/I initialization is stopped with message DLZ140I.

° Compatible definitions for active and alternate DL/I CICS

The active and the alternate DL/I CICS systems must have compatible DL/I and CICS resource definitions. There will not be any consistency check for DL/I or CICS resources such as CICS tables, DL/I nucleus (ACT, DBD, PSB, and so on).

10.3.2 Setup

To set up a DL/I CICS XRF environment, the SIT parameters XRF and the DLIOER must be coded as follows:

```
XRF=YES
DLIOER=ABEND|CONTINUE
```

For a detailed description on how to set up a CICS XRF environment, refer to the manual *CICS/VSE 2.1 XRF Guide*.

10.3.3 System Integrity

Depending on the online DL/I system configuration (logging specified), data integrity is ensured because alternate CICS together with DL/I performs backout during takeover processing. This applies also for DL/I MPS batch tasks and CICS ISC environments.

Until you repaired the database that has been stopped by an I/O error, automatic backout during CICS restart (EMER/TAKEOVER) will be suppressed for databases associated with any PSB that refers to that database. In this case, you have to perform a recovery of the damaged database by using the DL/I recovery utility.

------------ Diagnosis, Modification or Tuning Information -------------

Layout of Database Open Repaired Log Record (log id X'2F')

```
DLPOPEN  DSECT
DLOLEN   DS    H                   LENGTH OF OPEN RECORD
DLOSPACE DS    H'0'                ZERO
DLOCODE  DS    X                   RECORD TYPE CODE - X'2F'
         DS    X                   RESERVED
DLOORG   DS    X                   DATA SET ORGANIZATION
DLOESDS  EQU   X'00'               ESDS ORGANIZATION
DLOKSDS  EQU   X'04'               KSDS ORGANIZATION
DLOTORF  DS    CL1                 TYPE OF OPEN RECORD FLAG
```

```
     DLOIOERR EQU   X'80'               OPEN REPAIRED RECORD
              DS    XL4                 BINARY ZERO
     DLOCI    DS    XL2                 CONTROL INTERVAL LENGTH
              DS    XL2                 BINARY ZERO
     DLOFILE  DS    CL8                 DATA SET FILENAME (ACB)
     DLODMB   DS    CL8                 DMB NAME
     DLOACBNO DS    CL1                 DSG ACB NUMBER
     DLOHESDS EQU   2                   HISAM ESDS
     DLODATE  DS    CL3                 BINARY ZERO
     DLOTIME  DS    CL4                 BINARY ZERO
     DLOSEQNO DS    CL4                 LOG RECORD SEQUENCE NUMBER
     DLOEND   EQU   *
     DLPOLEN  EQU   *-DLPOPEN           LENGTH OF OPEN LOG RECORD
     DLOSEQ   DS    CL4                 LOG RECORD SEQUENCE NUMBER
     DLPOLENG EQU   *-DLPOPEN           LENGTH OF OPEN RECORD FOR
     *                                  LOG PRINT ROUTINE
```

Layout of Database I/O Error Stop Log Record (log id X'31')

```
     DLPIOERR DSECT
     DLILEN   DS    H                   LENGTH OF STOP RECORD
     DLISPACE DS    H'0'                ZERO
     DLICODE  DS    X                   RECORD TYPE CODE - X'31'
     DLISTOP  EQU   X'31'               STOP LOG RECORD
              DS    X                   RESERVED
     DLIORG   DS    X                   DATA SET ORGANIZATION
     DLIESDS  EQU   X'00'               ESDS ORGANIZATION
     DLIKSDS  EQU   X'04'               KSDS ORGANIZATION
     DLISTLRF DS    X                   STOP LOG RECORD FLAG
     DLIIOERR EQU   X'80'               STOPPED-IOERROR-RECORD
     DLIPSB   DS    CL8                 PSB NAME
     DLIFILE  DS    CL8                 DATA SET FILE NAME (ACB)
     DLIDMB   DS    CL8                 DMB NAME
     DLIACBNO DS    CL1                 DSG ACB NUMBER
     DLIHESDS EQU   2                   HISAM ESDS
     DLIDATE  DS    CL3                 BINARY ZERO
     DLITIME  DS    CL4                 BINARY ZERO
     DLISEQNO DS    CL4                 LOG RECORD SEQUENCE NUMBER
     DLIEND   EQU   *
     DLPILEN  EQU   *-DLPIOERR          LENGTH OF STOP LOG RECORD
     DLISEQ   DS    CL4                 LOG RECORD SEQUENCE NUMBER
     DLPILENG EQU   *-DLPIOERR          LENGTH OF STOP LOG RECORD
     *                                  FOR LOG PRINT ROUTINE
```

|--------- End of Diagnosis, Modification or Tuning Information ---------|

The DL/I Logprint Utility (DLZLOGP0): The DL/I Logprint Utility (DLZLOGP0) is enhanced to print out new log records. The printout of an open repaired record is as follows:

```
     OPEN REPAIRED RECORD, DSORG = (E)SDS (/HISAM)
                                  (K)
     SEQNO = xxxxxxxx, DMBNAME = aaaaaaaa, FILENAME = aaaaaaaa, REPAIRED
```

The printout of an I/O error stop record is as follows:

```
     I/O ERROR STOP RECORD, DSORG = (E)SDS (/HISAM)
```

```
                                              (K)
     SEQNO = xxxxxxxx, DMBNAME = aaaaaaaa, FILENAME = aaaaaaaa, PSBNAME = aaaaaaaa
```

where:

```
     xxxxxxxx = hexadecimal numbers
     aaaaaaaa = alphanumeric characters
```

Note: When specifying LO / LS control statements, remember that PSBname and DBDname are part of these log records.

11.0 Chapter 11. New Messages

```
 ___ In this Edition ... ______________________________________________
|                                                                       |
| The following messages are new for DL/I 1.10:                         |
|                                                                       |
|     DLZ1059                                                           |
|     DLZ1060                                                           |
|     DLZ109I                                                           |
|     DLZ379I                                                           |
|                                                                       |
| The following messages are included, because changes have been made   |
| for DL/I 1.10:                                                        |
|                                                                       |
|     DLZ1055                                                           |
|     DLZ002I                                                           |
|     DLZ015I                                                           |
|     DLZ037I                                                           |
|     DLZ058I                                                           |
|     DLZ096I                                                           |
|                                                                       |
| All other messages are new for DL/I 1.9.                              |
|                                                                       |
| For messages not listed in the following, refer to the manual DL/I    |
| Messages and Codes.                                                   |
|                                                                       |
|_______________________________________________________________________|
```

Subtopics:

- 11.1 DLZ1055
- 11.2 DLZ1059
- 11.3 DLZ1060
- 11.4 DLZ002I
- 11.5 DLZ015I
- 11.6 DLZ037I
- 11.7 DLZ050I
- 11.8 DLZ051I
- 11.9 DLZ053I
- 11.10 DLZ054I
- 11.11 DLZ058I
- 11.12 DLZ096I
- 11.13 DLZ107I
- 11.14 DLZ109I
- 11.15 DLZ140I
- 11.16 DLZ141I

- 11.17 DLZ142A
- 11.18 DLZ379I

11.1 DLZ1055

DLZ1055 REMOTE PSB WITH LOCAL COMPONENT NOT ALLOWED IN DFHMIRS TYPE=PROGRAM STATEMENT, ENTRY IGNORED IN DFHMIRS TYPE=PROGRAM STATEMENT

Explanation: Program DFHMIRS, the CICS/VSE intercommunication mirror program, is not allowed to schedule an RPSB with a local component. The name of this RPSB was specified in the TYPE=PROGRAM statement for DFHMIRS. Therefore, the RPSB entry in the DFHMIRS TYPE=PROGRAM statement will be ignored. Definition of this PSB as a remote PSB is allowed.

11.2 DLZ1059

DLZ1059 HSMODE = ***operand*** **INVALID, HSMODE = BELOW ASSUMED**

Explanation: The HSMODE operand of the DLZACT TYPE=CONFIG statement is incorrectly specified. The allowed HSMODE operands are ANY or BELOW.

The default HSMODE=BELOW will be assumed.

11.3 DLZ1060

DLZ1060 'DFHMIR' TYPE=PROGRAM STATEMENT CHANGED INTO 'DFHMIRS' TYPE=PROGRAM STATEMENT

Explanation: With CICS/VSE 2.2, the CICS intercommunication mirror program name has been changed from DFHMIR to DFHMIRS. However, DL/I 1.10 still accepts the old name DFHMIR in the TYPE=PROGRAM statement, but creates an entry for DFHMIRS in the Application Control Table (ACT) phase.

To avoid the MNOTE in the ACT generation, specify DFHMIRS in the TYPE=PROGRAM statement of the ACT source.

11.4 DLZ002I

DLZ002I BATCH DL/I ABNORMAL TERMINATION COMPLETE

Explanation: This message follows DLZ001I if the buffer pool records were successfully purged (written) and the data bases closed. A storage dump is produced.

The following provides information on the *storage dump* and the layout of the *save areas*.

------------ Diagnosis, Modification or Tuning Information -------------

Storage Dump -- The following should be noted from the storage dump:

Register	**Contents**
2	Addressable part of SCD (address of SCD plus 96 bytes).
3	Address of user save area if abnormal end was entered with a pseudo-abend. If entered for a program check or abnormal end, register 3 value is not set.
4	Contains original 2-byte error code at entry to pseudo-abend.
5	Address of STXIT AB save area if abend indicator shows that STXIT AB processing has been entered.
6	Address of the application program entry point.
7	Address of STXIT PC or pseudo-abend save area.
9	Address of PST.

Changes to Save Area Layout

As of DL/I 1.10, the save area has been extended to the *new* layout that is introduced with VSE/ESA 1.3. With DL/I 1.10:

° If STXIT PC or STXIT AB has been issued from within DL/I, the *new* layout will be used.
° If STXIT PC has been set from a PL/I application, the *old* layout is still valid.

Save Area Layout for STXIT PC

At label SCDABSAV in the SCD is a pointer to the STXIT or pseudo-abend save area which contains the following information:

Dec	**(Hex)**	**Length**	**Description**
-32	(-20)	32	Constant: ***** DL/I ABEND SAVE AREA *****

```
     0   (00)       8  Program old PSW (program check only; BC mode PSW)
     8   (08)      64  Registers 0 to 15
    72   (48)       8  EC mode PSW
    80   (50)       8  Reserved
    88   (58)      64  AB exit: Error information dependent on cancel code
   152   (98)      64  Save area for access registers 0 to 15
                       (not applicable for DL/I)
   216   (D8)      17  Constant: ABEND INDICATORS*
   233   (E9)       1  Abend reason indicator:
                         X'80' - Entered by STXIT AB'
                         X'40' - Entered by STXIT PC'
                         X'20' - Abend in progress'
                         X'10' - Buffer pool or data base damage'
                         X'08' - Buffer pool unload and close complete'
   234   (EA)       1  STXIT AB reason code as returned by the system
                       (low order byte of register 0)
   235   (EB)      25  Constant: *PSTFNCNT,RTCDE,OFFST-XXH
   260  (104)       1  PSTFNCTN (function code)
   261  (105)       1  PSTRTCDE (return code)
   262  (106)       2  PSTOFFST (offset)
   264  (108)      24  Constant: PSTBLKNM,BYTNM,DATA-FFF-
   288  (120)       4  PSTBLKNM (relative block number)
   292  (124)       4  PSTBYTNM (RBA or relative record number)
   296  (128)       4  PSTDATA (address of request data)
   300  (126)      24  Constant: PSTSV1 THROUGH PSTV3---
   324  (144)      72  PSTV1 (save area)
   396  (186)      72  PSTV2 (save area)
   468  (1D4)      72  PSTV3 (save area)
```

Save Area Layout for STXIT AB

If the abend indicator shows that STXIT AB processing has been entered, the STXIT AB save area may be located as follows:

1. Find the address of the SCD extension (label SCDEXTBA).
2. Find the address of the STXIT AB save area (label SCDEABSV).

This address points to the STXIT AB save area which contains the following information:

```
   Dec  (Hex)  Length  Description

     0   (00)       8  Program old PSW (BC mode PSW)
     8   (08)      64  Registers 0 to 15
    72   (48)       8  EC mode PSW
    80   (50)     136  Information not applicable for DL/I STXIT PC
```

The entry point address of the application program is in the SCDAPSTR field of the SCD. The DL/I high address rounded up to page boundary is in the field SCDDLIUP.

|--------- End of Diagnosis, Modification or Tuning Information ---------|

User Response: Take appropriate action.

Note: For information on additional aids for interpreting and debugging dumps, refer to the manual *DL/I Diagnostic Guide* under "Chapter: Interpreting And Debugging Dumps."

11.5 DLZ015I

DLZ015I PARAMETER STATEMENT DATA INVALID

Explanation: A DL/I parameter statement contains one of the following error conditions:

° Data did not start in column 1.

° A field length was invalid.

° A required field was omitted.

° An expected continuation statement was not found.

° A continuation statement was specified for SYSLOG input.

° The LOG parameter was specified a second time.

° The TRACE parameter was specified a second time.

° The HSMODE parameter was not specified as HSMODE=ANY or HSMODE=BELOW.

° ULR used without DLZURGU0.

° PLU used without DLZURGU0.

° PLU used and the specified PSB references a DBD or an index DBD.

User Response: Correct the error and execute the job again.

11.6 DLZ037I

DLZ037I DL/I STATUS=xx RETURNED, STMT 'yyyyyyyy', PROGRAM 'zzzzzzzz' TERMINATED

Explanation: A DL/I status code (xx) indicating an abnormal condition was returned to the application program (zzzzzzzz) and the program was abnormally terminated. See *DL/I Messages and Codes* under "DL/I Status Codes" for an explanation of the status codes. The yyyyyyyy is the statement number of the command that generated the status code, if available. For PL/I, the statement number will be taken from columns 73 and 80 of the listing produced by the CICS/VS translator. For DOS/VS COBOL and VS COBOL II, the statement number will be six characters taken from column 1 to 6.

User Response: Correct the error and rerun the application program.

11.7 DLZ050I

DLZ050I DEVICE MISMATCH FOUND IN DBD *dbdname*

Explanation: DEVICE=CKD was specified in DATASET statement of the database definition, but FBA was found in the device characteristics extracted during open of the data set. For HD/HDAM/HIDAM databases the value of the SCAN parameter is set to the default of FBA devices (32768). Processing continues and the return code is set to 4. For HSAM/SHSAM databases the mismatch leads to a termination. The return code is set to 12.

User Response: Correct the DEVICE parameter in the DATASET statement of DBDGEN for HSAM/SHSAM data bases and rerun the program. For HD/HDAM/HIDAM data bases the DEVICE parameter should be corrected if the SCAN parameter should not have the default value.

11.8 DLZ051I

DLZ051I GETVIS ERROR, RETURNCODE = *rc*

Explanation: The OPEN/CLOSE routine encountered an error when executing a GETVCE request. The program terminated. The return code is given in decimal format. For an explanation of the GETVCE return code *rc*, refer to the manual *VSE/ESA Messages and Codes*, in the section "VSE/Advanced Functions Codes and SVC Errors."

User Response: Correct the error and rerun the program.

11.9 DLZ053I

DLZ053I DL/I INITIALIZATION COMPLETE [FOR XRF STANDBY | FOR XRF TAKEOVER]

Explanation: DL/I online initialization was successful and no errors were detected.

The message "DLZ053I DL/I INITIALIZATION COMPLETE" is issued within an active CICS XRF environment or in a CICS system without XRF.

The message "DLZ053I...FOR XRF STANDBY" and "DLZ053I...FOR XRF TAKEOVER" is issued in an alternate CICS XRF environment.

The DL/I initialization in the alternate CICS XRF environment is a two-stage process. 'STANDBY' indicates that the alternate DL/I initialization has completed but without open of the data bases. DL/I is waiting in a standby mode and is ready to take over if the active DL/I CICS system terminates due to an error situation or a console command. 'TAKEOVER' indicates that the alternate DL/I initialization has completed including the open of the data bases and takeover has finished.

User Response: None required.

11.10 DLZ054I

DLZ054I DL/I INITIALIZATION ERROR(S) DETECTED [FOR XRF STANDBY | FOR XRF TAKEOVER]

Explanation: DL/I online initialization was successful, but one or more errors were detected.

The message "DLZ054I DL/I INITIALIZATION ERROR(S) DETECTED" is issued within an active CICS XRF environment or in a CICS system without XRF.

The message "DLZ054I...FOR XRF STANDBY" and "DLZ054I...FOR XRF TAKEOVER" is issued in an alternate CICS XRF environment.

The DL/I initialization in the alternate XRF environment is a two-stage process. 'STANDBY' indicates that the alternate DL/I initialization has completed but without open of the data bases. DL/I is waiting in a standby mode and is ready to take over if the active DL/I CICS system terminates due to an error situation or a console command. 'TAKEOVER' indicates that the alternate DL/I initialization has completed including the open of the data bases and takeover has finished. Note that the error indication from first stage initialization is propagated.

User Response: None required.

11.11 DLZ058I

DLZ058I INSUFFICIENT STORAGE TO START DL/I

Explanation: The virtual storage area is too small to hold the

DL/I modules or buffer pools.

User Response:

For Batch Environment

This message is followed by an immediate abnormal termination and dump.

Check the following:

- The HDBFR parameter in the DL/I parameter statement can be used to decrease the number of buffers required. It is possible that the parameter is not read because of a missing continuation character in column 72.

  If this caused the problem, correct the DL/I parameter statement and run the job again, or proceed as shown below under *Corrective Action*.

- The eligible DL/I action modules have not been loaded in the shared virtual area (SVA). (For a complete list of SVA-eligible DL/I phases, refer to "SVA Loading for DL/I" in topic 5.1.)

  If this caused the problem, install the DL/I action modules in the SVA and run the job again. Alternatively, you can proceed as shown below under *Corrective Action*.

- There is insufficient program space in the program area to load the application. If this is the case, proceed as shown below under *Corrective Action*.

Corrective Action:

- Increase the SIZE parameter in the EXEC statement and resubmit the job,

  or
- Increase the partition size and the SIZE parameter in the EXEC statement and resubmit the job.

For Online Environment

If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue only with CICS initialization, enter either GO or RUN.

DL/I will not be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I functions.

- To stop DL/I and CICS initialization, enter either:

  CANCEL to end DL/I and CICS initialization.

  DUMP to end DL/I and CICS initialization, and to dump the contents of storage at this point.

To correct this problem, check that:

- The HDBFR parameter in the DLZACT TYPE=BUFFER statement can be used to decrease the number of buffers required. It is possible that the parameter has no effect because of an error in the DLZACT statements that were used to build this nucleus.

  If this caused the problem, correct the DLZACT TYPE=BUFFER statements and rebuild the DL/I nucleus (ACT) by using the ACTGEN procedure, or proceed as shown below under *Corrective Action*.

- The eligible DL/I action modules have not been loaded in the shared virtual area (SVA). (For a complete list of SVA-eligible DL/I phases, refer to "SVA Loading for DL/I" in topic 5.1.)

  If this caused the problem, install the DL/I action modules in the SVA, or proceed as shown below under *Corrective Action*.

Corrective Action:

- Increase the SIZE parameter in the EXEC statement and rerun the CICS/VS and DL/I initialization job,

  or
- Increase the partition size and the SIZE parameter in the EXEC statement and resubmit the job.

11.12 DLZ096I

DLZ096I STXIT {AB|PC} ENTERED, MPS BATCH PARTITION ENDED ABNORMALLY

Explanation: Control was passed to the STXIT AB or STXIT PC entry point in the MPS batch abend handler routine (DLZMABND).

------------ Diagnosis, Modification or Tuning Information -------------

User Response: Examine the log and dump (if printed), to determine the cause of the error. The corresponding register save areas are immediately preceded in the dump of module DLZMPI00 by the eye-catchers 'AB SAVE' or 'PC SAVE' respectively. The one-byte reason code for the AB STXIT entry is preceded by 'AB REASON CODE'. In the dump, Register 3 contains the address of the applicable save area.

Note:

As of DL/I 1.10, the save area has been extended to the *new* layout that is introduced with VSE/ESA 1.3. With DL/I 1.10:

° If STXIT PC or STXIT AB has been issued from within DL/I, the *new* layout will be used.
° If STXIT PC has been set from a PL/I application, the *old* layout is still valid.

|--------- End of Diagnosis, Modification or Tuning Information ---------|

For information on additional aids for interpreting and debugging dumps, refer to the manual *DL/I Logic Extensions* under "Diagnostic Aids for MPS Messages."

11.13 DLZ107I

DLZ107I RECORD LENGTH (*nnnnn*) EXCEEDS DEVICE LIMIT

Explanation: The specified record size for HSAM/SHSAM data bases exceeds the maximum record length:

```
track capacity for CKD devices,
32766 bytes    for FBA devices.
```

User Response: Shorten the segment or split it into one or more smaller segments with a parent/child relation, or change the device type, or correct your record length specification.

11.14 DLZ109I

DLZ109I DL/I PARAMETERS ABOVE 16MB, TASK TERMINATED

Explanation: DL/I has detected that parameters and/or the parameter list are located above the 16MB line of storage. (Parameters and parameter list must be located below the 16MB line of storage.)

User Response: Ensure that the parameters and the parameter list passed to DL/I are located below the 16MB line of storage.

For information relating to:

- High level assembler applications, refer to "Placing DL/I Parameter Definitions Below the 16MB Line of Storage" in topic 9.1.7.
- VS COBOL II applications, refer to "Creating COBOL II Applications with 31-Bit Capability" in topic 9.1.5. Specify the compile option DATA(24).

11.15 DLZ140I

DLZ140I CICS JOURNAL NOT ACTIVE BUT REQUIRED FOR XRF SUPPORT

Explanation: A DL/I CICS XRF environment requires CICS journaling for DL/I logging. This message is issued if you run CICS without journaling or if CICS journaling was suppressed by UPSI bits 6 and 7 in the CICS initialization job. The message DLZ140I may be followed by message DLZ040A.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message. Enter CANCEL as response to message DLZ140I to terminate the DL/I and CICS initialization. In order to enter other possible replies, refer to message DLZ049I. In order to correct the problem, set UPSI bits 6 and 7 in CICS initialization job to 0 to activate CICS journaling. Rerun the job.

11.16 DLZ141I

DLZ141I I/O ERROR HAS OCCURRED IN DATABASE *dbdname*

Explanation: During CICS warm start, emergency restart or takeover the DL/I recovery routines found a database stop record in the CICS restart data set indicating that an I/O error had occurred for the database *dbdname*. Message DLZ004I or DLZ005I was previously issued. Message DLZ141I is followed by DLZ142A. Batch recovery actions have to be taken to physically repair the database.

For more information, refer to the message DLZ142A.

User Response: None required.

11.17 DLZ142A

DLZ142A ENTER CONTINUE, IGNORE OR CANCEL

Explanation: Message DLZ142A follows DLZ141I.

User Response: Enter one of the following:

CONTINUE to indicate that the database has already been repaired using the DL/I recovery utility.

Automatic backout will be performed for any "inflight" log records that are found on the system log. This (in addition to previous executed batch forward recovery) ensures database integrity.

DL/I initialization continues.

Note: If OPEN=DEFERRED is coded in File Control Table (FCT), the database must be opened using the DL/I system call STRT before the database can be used again.

IGNORE to indicate that the database has not been repaired. The indicated database remains closed and cannot be accessed.

Therefore, automatic backout cannot be performed during CICS start or restart for any database that is associated with the PSB notified in message DFH5723 (issued by CICS).

DL/I initialization continues.

CANCEL to end DL/I and CICS processing. DL/I and CICS initialization terminates.

11.18 DLZ379I

DLZ379I ERROR - INTERNAL WORKING STORAGE HAS BEEN EXCEEDED

Explanation: The *change accumulation* or *recovery* utility has detected that one of the internal work areas has been exceeded. The program terminates and issues the message DLZ385I.

The size of these (GETVIS) work areas depends mainly on the CI size of the processed data base. Also consider that there may be other CUMIN or CUMOUT records to be processed that require work areas larger than the standard size.

The work areas and their standard sizes are as follows:

- *Recovery Utility* (DLZURDB0)

  CUMIN input work area size:

      CI size x 2.5, or
      63K if MAXI is specified in S-card.

  The minimum is 10K, the maximum is 63K.

- *Change Accumulation Utility* (DLZUCUM0)

  - CI work area size (if CUMIN provided):

        CI size from first log record, or (if greater)
        nnK CI size specified in ID-card (nn = 04 ... 30),
        or
        30K if MAXI is specified in ID-card.

    The minimum is 4K.

  - CUMIN input work area size:

        (Previous) CI work area size x 2.5.

    The minimum is 10K, the maximum is 63K.

  - CUMOUT output work area size:

        (Previous) CI work area size x 2.5.

    The minimum is 10K, the maximum is 63K.

User Response: In the applicable utility:

1. Specify MAXI in column 25-28 of the S- or ID-card (control statement).

2. Resubmit the job.

A.0 Appendix. Summary of Customer Interfacs

This appendix contains General-use Programming Interface and Associated Guidance Information, and Product-sensitive Programming Interface and Associated Guidance Information.

Subtopics:

• A.1 DL/I Macros Intended for Customer Use

A.1 DL/I Macros Intended for Customer Use

Figure 32 identifies the DL/I macros that are provided to allow a customer installation to write programs that use the services of DL/I. Only the macros identified in the figure should be used to request or receive the services of DL/I.

Note: All macros mentioned in this appendix are distributed in system library PRD2.DBASE.

Macro Name	General Use	Product Sensitive	Described in (See Legend, below)
ACCESS	x		RDU, IRDU, GNU
CALLDLI	x		DIAG, CALL
DATASET	x		RDU, IRDU, GNU
DBD	x		RDU, IRDU, GNU, DIAG
DBDGEN	x		RDU, IRDU, GNU, DIAG
DLZACT	x		RDU, IRDU, GNU, DIAG, DBA
DLZBFFR		x	DIAG
DLZBFPL		x	DIAG, DBA
DLZCTRL		x	RDU, RELG
DLZDIB	x		CALL
DLZHDC10-40		x	DBA
DLZMDLI		x	DBA
DLZNN	x		LLC
DLZNNICT	x		LLC
DLZSLC	x		RDU
DLZSTBF		x	RELG
DLZTRACE	x		DIAG
DLZTXITO		x	DIAG
DLZUIB	x		CALL
FIELD	x		RDU, IRDU, GNU
FINISH	x		RDU, IRDU, GNU
LCHILD	x		RDU, IRDU, GNU
PCB	x		RDU, IRDU, GNU
PSBGEN	x		RDU, IRDU, GNU
SEGM	x		RDU, IRDU, GNU
SENFLD	x		RDU, IRDU, GNU
SENSEG	x		RDU, IRDU, GNU
VIRFLD	x		RDU, IRDU, GNU
XDFLD	x		RDU, IRDU, GNU

Figure 32. Summary of Customer Interfaces

Legend for the above table:

Abbreviation	Publication
CALL	*DL/I Application Programming: CALL and RQDLI Interfaces*
DBA	*DL/I Data Base Administration*
DIAG	*DL/I Diagnostic Guide*
GNU	*DL/I Guide for New Users*

HLPI	*DL/I Application Programming: High Level Programming Interface*
IRDU	*DL/I Interactive Resource Definition and Utilities*
RELG	*DL/I Release Guide*
RDU	*DL/I Resource Definition and Utilities*

BIBLIOGRAPHY Related IBM Manuals

Subtopics:

- BIBLIOGRAPHY.1 IBM DL/I DOS/VS
- BIBLIOGRAPHY.2 IBM VSE/ESA
- BIBLIOGRAPHY.3 IBM VSE/SP Version 3
- BIBLIOGRAPHY.4 IBM VSE/SP Version 4
- BIBLIOGRAPHY.5 IBM VSE/ICCF
- BIBLIOGRAPHY.6 IBM CICS
- BIBLIOGRAPHY.7 IBM SQL/DS
- BIBLIOGRAPHY.8 IBM ISPF
- BIBLIOGRAPHY.9 IBM VS COBOL II
- BIBLIOGRAPHY.10 IBM High Level Assembler

BIBLIOGRAPHY.1 IBM DL/I DOS/VS

Figure 33 lists *all* manuals within the DL/I DOS/VS library. For assistance in locating information in the library, refer to the manual *DL/I Library Guide and Master Index*.

In the figure:

° The "dash number" (for example, the **-1** in GH24-5008**-1**) shows which version of the manual applies for DL/I DOS/VS 1.10.

° The "(1)" means that the manual is referenced in this manual.

Form Number		Title
GH20-1246-9		DL/I General Information
GH24-5008-1		DL/I Library Guide and Master Index
GH24-5031-4		DL/I Licensed Program Specifications
SC33-6211-4	(1)	DL/I Release Guide
SH24-5001-4	(1)	DL/I Guide for New Users
SH24-5022-1		DL/I Application and Data Base Design
SH24-5011-1	(1)	DL/I Data Base Administration
SH24-5021-2	(1)	DL/I Resource Definition and Utilities
SH24-5029-0	(1)	DL/I Interactive Resource Definition and Utilities
SH24-5009-2	(1)	DL/I Application Programming: High Level Programming Interface
SH12-5411-6	(1)	DL/I Application Programming: CALL and RQDLI Interfaces
SH12-5414-10	(1)	DL/I Messages and Codes

```
     SH24-5002-4  (1)    DL/I Diagnostic Guide
     SH24-5030-0         DL/I Recover and Restart Guide


     SH20-9046-3         IBM System/370 LLC/CC in DL/I DOS/VS
                         Program Reference and Operations Manual


     Summaries:


       SX24-5103-4       DL/I Reference Summary: Application Programming
       SX24-5104-4       DL/I Reference Summary: System Programming
       SX24-5120-2       DL/I Reference Summary: High Level Programming Interface


     Logic Manuals:


       LY12-5016-7       DL/I Logic, Volume 1
       LY12-5015-1       DL/I Logic, Volume 2
       LY33-9123-1 (1)   DL/I Logic Extensions
```

Figure 33. Manuals in the DL/I DOS/VS Library

BIBLIOGRAPHY.2 IBM VSE/ESA

VSE/ESA Extended Addressability, SC33-6524
VSE/ESA Planning, SC33-6503
VSE/ESA Installation and Service, SC33-6504
VSE/ESA Messages and Codes, SC33-6507

BIBLIOGRAPHY.3 IBM VSE/SP Version 3

VSE/SP Installation Version 3, SC33-6305
VSE/SP Messages and Codes Version 3, SC33-6310

BIBLIOGRAPHY.4 IBM VSE/SP Version 4

VSE/SP Installation Version 4, SC33-6404
VSE/SP Messages and Codes Version 4, SC33-6407

BIBLIOGRAPHY.5 IBM VSE/ICCF

VSE/Interactive Computing and Control Facility User's Guide, SC33-6563

BIBLIOGRAPHY.6 IBM CICS

CICS/VSE 2.2 Release Guide, &bc22rgn.

CICS/VSE 2.2 System Definition and Operations Guide, SC33-0706

&bcsdopx., SC33-0706

CICS/VSE 2.1 XRF Guide, SC33-0704

CICS/DOS/VS 1.7 Installation and Operations Guide, SC33-0070

BIBLIOGRAPHY.7 IBM SQL/DS

SQL/Data System Administration for VSE Version 3.2, GH09-8096

BIBLIOGRAPHY.8 IBM ISPF

Interactive System Productivity Facility (ISPF) Dialog Management Services, SC34-2088

BIBLIOGRAPHY.9 IBM VS COBOL II

Migration

VS COBOL II Migration Guide for VSE, GC26-3150

VSE/ESA DOS/VS COBOL to VS COBOL II Migration Considerations, GG24-3791

Installation

VS COBOL II Installation and Customization for VSE, SC26-4696

Application Programming

VS COBOL II Application Programming Guide for VSE, SC26-4697

VS COBOL II Application Programming Language Reference, GC26-4047

BIBLIOGRAPHY.10 IBM High Level Assembler

High Level Assembler Programmer's Guide, SC26-4941

High Level Assembler Language Guide, SC26-4940

GLOSSARY Glossary

Listed below are the common terms, abbreviations, and acronyms used throughout this document with a brief definition and/or explanation.

Numerals

31-bit addressing. Provides addressability for address spaces of up to 2 gigabytes.

A

ACB. DL/I application control block, created by the output of DBDGEN and PSBGEN.

ACT. DL/I application control table.

addressing mode (AMODE). A program attribute that refers to the address length that a program is prepared to handle on entry. Addresses may be either 24 bits or 31 bits in length. In 24-bit addressing mode, the processor treats all virtual addresses as 24-bit values; in 31-bit addressing mode, the processor treats all virtual addresses as 31-bit values.

AMODE. See *addressing mode*.

APAR. Authorized Programming Analysis Report.

C

CICS*. Customer Information Control System, a licensed IBM program for online environments.

CLC. Component level code.

CMS. Conversational Monitoring System.

Configuration. The combined hardware and software products that make up a data processing system.

CPU. Central processing unit of the computer hardware system.

D

DA. DL/I Documentation Aid.

DBD. Database description.

DB/DC. Database/data communication.

DL/I. Data Language/One, a licensed IBM program.

F

FBA disk device. A fixed block architecture storage device for data in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the file.

FCT. CICS file control table.

G

GSCD. Get system contents directory.

GSTA. Get statistics.

H

HD. Hierarchical direct, a DL/I access method.

HDAM. Hierarchical Direct Access Method.

HIDAM. Hierarchical Indexed Direct Access Method.

HISAM. Hierarchical Indirect Sequential Access Method.

HLPI. High Level Programming Interface, a DL/I function that allows the DOS/VS COBOL and PL/I Optimizer application programmer to process DL/I databases in batch, MPS batch, and CICS online environments.

HSAM. Hierarchical sequential access method, a DL/I access method.

I

IMF. Interactive Macro Facility, a DL/I function that allows the user to generate DBDs, PSBs, etc. from menu-driven display panels.

IMS/VS. Information Management System/Virtual Storage.

IPF. Interactive Productivity Facility.

IPL. Initial program load.

ISPF. Interactive System Productivity Function, a licensed IBM program required for the use of DL/I IMF and IUG functions. It is the dialog manager for interactive applications.

ISQL. Interactive Structured Query Language.

IUG. Interactive Utilities Generation, a DL/I function that allows the user to generate utility job streams from menu driven display panels.

J

JCL. job control language

M

MPS. Multiple Partition Support.

MSHP. Maintain System History Program.

P

PCB. Program communication block.

PCT. CICS program control table.

PPT. CICS processing program table.

PSB. Program specification block.

PST. Partition specification table.

PTF. Program temporary fix.

R

residency mode (RMODE). A program attribute that refers to the location where a program is expected to reside in virtual storage.

RMODE. See *residency mode*.

S

SDL. System directory list.

SHSAM. Simple Hierarchical Sequential Access Method.

SLC. Storage Layout Control table; for use in an online environment to specify in which sequence DL/I phases are to be loaded from the library during DL/I initialization.

SPE PTF. Small program enhancement PTF.

SQL/DS*. Structured Query Language/Data System.

SVA. Shared virtual area, located in the highest address range of virtual storage. It can contain a system directory list (SDL) of often used phases, resident programs that can be shared between partitions, and an area for dynamic allocation to components of VSE.

System History File. Part of the IBM-distributed VSE system and maintained under the file name IJSYSHF on the (preferred) logical unit SYSREC. This file normally contains all system history status information (product numbers, component IDs, CLC numbers, PTF and APAR numbers, and so on) and is updated by MSHP.

U

UPSI. User program switch indicator.

V

virtual disk. A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as workspace. Although the virtual disk exists in storage, it appears as a real FBA disk device to the user program. All I/O operations directed to a virtual disk are intercepted and the data to be written to, or read from, the disk is moved to or from a data space.

Like a data space, a virtual disk can hold only user data; it does not contain shared areas, system data or programs. Unlike an address space or a data space, data is not directly addressable on a virtual disk. To manipulate data on a virtual disk, the program has to perform I/O operations.

VM/SP. Virtual Machine/System Product.

VS. Virtual Storage.

VSE. Virtual storage extended: A system that consists of a basic operating system (VSE/Advanced Functions) and any IBM supplied and user-written programs required to meet the data processing needs of a user. Its current version is called VSE/ESA.

VSE/ESA*. VSE/Enterprise Systems Architecture.

VSE/VSAM. VSE/Virtual Storage Access Method: the main access method, for direct or sequential processing of fixed and variable length records on disks.

X

XPCC. Cross-partition communication control.

COMMENTS Readers' Comments

```
IBM Data Language/I
Disk Operating System/Virtual Storage
Release Guide
Version 1 Release 10


Publication No. SC33-6211-04





We would appreciate any comments you may have about this publication, 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.  Feel free to comment on technical accuracy, retrievability,
clarity, or overall structure of the publication.
```

If you wish a reply, give your name and address below. No postage stamp is necessary if mailed in the USA. Elsewhere, you may mail directly to:

IBM Deutschland GmbH
Department 3248
Schoenaicher Strasse 220
D-7030 Boeblingen
Federal Republic of Germany

You may also respond via **FAX** to: Department 3248, Country Code 49, (0)7031-16-3456

Name	______
Company or Organization	______
Address	______
	______
	______
Phone No.	______

IBM Library Server Print Preview

DOCNUM = SC33-6211-04
DATETIME = 03/12/93 11:09:15
BLDVERS = 1.2
TITLE = IBM DL/I DOS/VS Release Guide
AUTHOR =
COPYR = © Copyright IBM Corp. 1984, 1993
PATH = /home/webapps/epubs/htdocs/book