IBM Data Language/I
Virtual Storage Extended

IBM

Release Guide

Version 1 Release 11

IBM Data Language/I
Virtual Storage Extended

Release Guide

Version 1 Release 11

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Seventh Edition (January 2006)

This edition applies to Version 1 Release 11 of IBM Data Language/I Virtual Storage Extended (DL/I VSE), Program Number (5746-XX1),

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 Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
```

You may also send your comments by FAX or via the Internet:

```
Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456
```

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

Figures

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 expressly designated by IBM, are 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 Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Deutschland GmbH
Department 0790
Pascalstr. 100
70569 Stuttgart
Germany
```

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Programming Interface Information

This manual also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DL/I VSE. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface Information

End of Programming Interface Information

Trademarks and Service Marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

```
AT
CICS
CICS/ESA
CICS/MVS
CICS/VSE
DATABASE 2
DB2
DFSORT
IBM
IMS
NetView
OS/2
S/390
SP
SQL/DS
System/370
System/390
VisualLift
VM/ESA
VM/SP
VM/XA
VSE/ESA
VTAM
z/VSE
```

Year 2000 Ready

This IBM product, when used in accordance with its associated documentation, is capable of correctly processing, providing, and/or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with this IBM product properly exchange accurate date data with it.

About This Book

This book describes the latest enhancements provided for DL/I VSE 1.11 available since VSE/ESA 2.4.

It also includes information on installation, migration, customization, and other post-installation tasks. Changed and new messages are included as well.

The manual still includes information on earlier DL/I releases and environments for reference purposes.

Who Should Use This Book

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

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

How to Use This Book

Read first Chapter 1, "DL/I Release News," on page 1 to become familiar with the latest enhancements and functions provided by DL/I VSE 1.11. The chapter also includes migration information.

For an *overview* on features of earlier releases of DL/I, refer to Chapter 8, "Features Introduced with Earlier Releases of DL/I," on page 49.

Where to Find More Information

For a list of all DL/I publications available, refer to "Related IBM Manuals" on page 87.

---
z/VSE Home Page

z/VSE has a home page on the World Wide Web, which offers up-to-date information about VSE-related products and services, new z/VSE functions, and other items of interest to VSE users.

You can find the z/VSE home page at:

`http://www.ibm.com/servers/eserver/zseries/zvse/`

Summary of Changes for DL/I

The following summarizes the changes and enhancements provided with the various DL/I releases.

DL/I VSE 1.11 Enhancements

Refer to "DL/I VSE 1.11 Enhancements" on page 1 for details.

DL/I DOS/VS 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)

DL/I DOS/VS 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

DL/I DOS/VS 1.8 Enhancements

- Device Independence (APAR PL48345)
- Automatic Verification for DL/I ESDS Data Sets (APAR PL20988)
- Program Isolation Enhancement (APAR PL55587)

Chapter 1. DL/I Release News

Important Information

Starting with VSE/ESA 2.4, two versions of the DL/I optional program are available for use with VSE/ESA or z/VSE:

> **DL/I VSE 1.11**
>
> DL/I DOS/VS 1.10

DL/I VSE 1.11 is for use with a new CICS program, called the **CICS Transaction Server for VSE/ESA 1.1**. It replaces CICS/VSE 2.3.

Since you can still install CICS/VSE 2.3 in addition to the CICS Transaction Server, VSE/ESA and z/VSE allow you to run two different CICS-DL/I environments at the same time:

* DL/I VSE 1.11 together with the CICS Transaction Server 1.1
* DL/I DOS/VS 1.10 together with CICS/VSE 2.3

The manuals *VSE/ESA Planning* and *z/VSE Planning* provide further details about environments that can coexist under VSE/ESA or z/VSE.

DL/I VSE 1.11 Enhancements

The following enhancements are available with DL/I VSE 1.11.

Multiple MPS Systems

The DL/I MPS Master Partition Controller DLZMPC00 can be activated in more than one CICS/DLI partition at the same time. An MPS batch job can select any active CICS/DLI online system and process databases defined there (one connection per job step). The selection is done through new parameters of the DLI/DLR statement submitted with the MPS batch job.

Each MPS Master Partition Controller DLZMPC00 is started or stopped in its own partition and in the same way as before through the DL/I MPS Start/Stop transactions CSDA/CSDD.

New parameters for DLI/DLR statement:

The new parameters of the DLI/DLR statement are shown below. Either one or both of them, separated by a comma, can be entered:

```
        PARTID=xx
```

denotes the CICS/DLI MPS partition, where the database(s) to be processed are defined.

```
        APPLID=yyyyyyyy
```

denotes the CICS (generic) applid of the CICS/DLI MPS partition, where the database(s) to be processed are defined.

```
DLI,progname,psbname,PARTID=F2
DLI,progname,psbname,APPLID=DBDCCICS,RC=YES
DLR,progname,psbname,PARTID=F4,APPLID=PRODCICS
```

Figure 1. Examples of the PARTID and APPLID Parameters

The RC parameter can be placed anywhere after the psbname, together with or without the new PARTID/APPLID parameters.

If only one CICS/DLI MPS system is up, the new parameters are not required and the MPS batch jobs run as before.

Invocation through DLZCTRL

It is also possible to use the DLZCTRL program to dynamically test, if a batch job should run in the normal DL/I batch environment (DLZRRC00) or under the control of MPS (DLZMPI00). DLZCTRL has been introduced on pages 252/253 of the manual DL/I Resource Definition and Utilities (SH24-5021-02).

In order to let DLZCTRL make the correct decision a PARM value as shown below should be added to the EXEC DLZCTRL statement:

`// EXEC DLZCTRL,SIZE=1024K,PARM='PARTID=F4,APPLID=        '`

The following syntax rules apply to the PARM value:
- It must have a fixed length of 25 characters.
- Columns 1-7 must contain the string PARTID=.
- Columns 11-17 must contain the string APPLID=.
- At least one parameter should be specified.

DLZCTRL will call either DLZMPI00 or DLZRRC00 according to the following rules:
1. **PARM value present with correct syntax:** DLZCTRL will search for the CICS/DLI MPS system specified in the PARM value. If the requested CICS/DLI MPS system is active, DLZCTRL will call DLZMPI00, else it will call DLZRRC00.

   When specifying a PARM value, it is important that the DLI/DLR statement for DLZMPI00 contains matching PARTID/APPLID parameters. The PARM value is only used to test if a requested CICS/DLI MPS system is up (pre-check). The actual connection to send data between the MPS batch partition (DLZMPI00) and the CICS/DLI MPS system (DLZMPC00/DLZBPC00) is established from the PARTID/APPLID parameters in the DLI/DLR statement.
2. **PARM value present with incorrect syntax:** DLZCTRL will issue an error message and terminate.
3. **PARM value not present:** DLZCTRL will test, if **any** CICS/DLI MPS system is active. If yes, DLZCTRL will call DLZMPI00 else it will call DLZRRC00.

   This does not ensure that the CICS/DLI MPS system requested through a succeeding DLI/DLR statement is active. An unsuccessful MPS batch execution may follow (DLZ089I) instead of a successful regular batch execution. Therefore a PARM value should always be added.

> **Note:** The batch initialization program DLZRRC00 will disregard any PARTID/APPLID parameters.

More than 4 GB for HD Databases

DL/I HD (HDAM/HIDAM) databases can be allocated on more than one VSAM ESDS dataset. Up to five datasets can be defined. This increases the maximum database size to 20 GB (5 times 4 GB). The enhancement is transparent to user applications. The split into multiple datasets will be realized through an updated DBD definition and a reorganization or new initial load of the database:

Definition:

Each VSAM dataset is represented through a DATASET macro statement in the DBD-generation job. The database segments are allocated on the VSAM datasets according to the following rules:

- The first DATASET statement is entered with all parameters as before. It determines, on which VSAM dataset the root segment resides and all subsequently defined segments up to the second DATASET statement.

- The second DATASET statement denotes the VSAM dataset, on which all subsequently defined segments are stored. Any following DATASET statements and segment definitions are processed accordingly. Currently up to five DATASET statements are allowed.

- The second to the fifth DATASET macro statement have the following syntax:

```
        label     DATASET     DD1=filename
```

label   Can be defined on column 1, it has no effect.

filename   Denotes the 1-7 character VSAM symbolic filename.

Other parameters, which can be entered on the first DATASET statement are not supported. The DEVICE, BLOCK, SCAN and FRSPC values are taken from the first DATASET statement. This implies that all VSAM datasets must reside on the same DASD-type and have the same blocksize. Dataset definition parameters (DEFINE CLUSTER job) of the VSAM datasets should correspond.

Figure 2 on page 5 shows an example of a DBD definition, which allocates the database STDCDBP on four VSAM datasets: The root segment STSCCST and all following segments defined up to segment STSCLOC are allocated on the primary dataset STDCDBC. Segment STPCORD and any segments defined after it are allocated on the second dataset ST2CDBC. Starting with segment STCCITM, all segments are allocated on the third dataset ST3CDBC. The segments STSCSTA and STSCHIS will reside on the fourth dataset ST4CDBC.

There are no restrictions regarding the definition of logical relationships or secondary indexing.

Database Load: The database can be split into multiple VSAM datasets in the following way:

1. **Initial Load through user application:**

   The DBD generation job of the affected database has to be implemented according to the new definition rules (DBDGEN with multiple DATASET statements). PCB definitions (PSBGEN) have not changed. The new DBD and any new or existing PSB execution blocks have to be re-created by running the DL/I Block Builder utility DLZUACB0.PSB execution blocks have to be re-built

for all PSBs, which allow access to the affected database. After that a user application can load the database using the new control blocks.

Note: The facility to load a database stepwise via separate load runs (Partial Database Load) is not supported for databases residing on multiple datasets.

2. **Reorganization:**

The database has to be unloaded using the DL/I Unload Utility DLZURGU0 and the DBD definition in the existing format (one DATASET statement). After that, new DBD and PSB execution blocks have to be created as described above. Then the database can be reloaded into the new format using the DL/I Reload Utility DLZURGL0 and the new control blocks built.

The reorganization utilities for the creation of logical relationships and secondary indexes DLZURPR0, DLZURG10 and DLZURGP0 run as before.

Note: The Partial Database Reorganization utility (DLZPRCT1/2) is not supported for databases residing on multiple datasets.

Execution:

The allocation of a database on multiple datasets is transparent to applications. User programs do not have to be changed. All DL/I calls can be issued as before.

Additional DLBL statements are needed for all secondary datasets in DL/I batch and CICS/DLI startup jobs.

Utilities:

The DL/I utilities as documented in the DL/I manual *Resource Definition and Utilities* (SH24-5021) can be used in the same way, when databases are allocated on multiple datasets. There are no new or changed control statements. However, these changes in existing job streams are necessary:

- Additional DLBL statements must be supplied for all secondary datasets, whenever a utility directly accesses a database.
- When using the DL/I Image Copy utility DLZUDMP0, a 'D' control statement must be entered for each dataset (cols 13-19), which has been defined for the database. The control statements can be submitted with one job step.
- When using the DL/I Recovery utility DLZURDB0, each dataset of the database has to to be recovered in a separate job step. The dataset names are entered on the 'S' control statement (cols 13-19) as before.

```
           DBD                                                    X
                 NAME=STDCDBP,           DATA BASE DESCRIPTION NAME    X
                 ACCESS=HDAM,            HIERARCHICAL DIRECT           X
                 RMNAME=(DLZHDC10,       RANDOMIZING ROUTINE PHASENAME X
                 3,                      ROOT ANCHOR POINTS PER BLOCK  X
                 100,                    ROOT ADDR. AREA HI RELATIVE BLK X
                 600)                    INSERT BYTES LIMIT FOR RAA
           DATASET                                                 X
                 DD1=STDCDBC,           DLBL FILE NAME - FIRST DATASET X
                 DEVICE=CKD,             DISK DEVICE                   X
                 BLOCK=(4096),           VSAM CONTROL INTERVAL SIZE    X
                 SCAN=2                  # CYLINDERS SCAN FOR ISRT SPACE
           SEGM                                                    X
                 NAME=STSCCST,           SEGMENT NAME FOR CUST NAME/ADDR X
                 PARENT=0,               IT IS A ROOT SEGMENT          X
                 BYTES=106,              DATA LENGTH                   X
                 POINTER=TWIN            PHYSICAL TWIN FWD ONLY
           FIELD                                                   X
                 NAME=(STQCCNO,SEQ,U),   UNIQUE KEY FIELD  (CUST #)    X
                 BYTES=6,                FIELD LENGTH                  X
                 START=1,                WHERE IT STARTS IN SEGMENT    X
                 TYPE=C                  ALPHAMERIC DATA
                 .
                 .
                 .
           SEGM                                                    X
                 NAME=STSCLOC,           SEGMENT NAME CUSTOMER LOCATION X
                 PARENT=STSCCST,         PARENT IS CUST. NAME/ADDR SEGM X
                 BYTES=106,              FIELD LENGTH                  X
                 POINTER=TWINBWD         BOTH PHYS. TWIN FWD AND BWD
           FIELD                                                   X
                 NAME=(STQCLNO,SEQ,U),   UNIQUE KEY FIELD (LOCATION #) X
                 BYTES=6,                FIELD LENGTH                  X
                 START=1,                WHERE IT STARTS IN SEGMENT    X
                 TYPE=C                  ALPHAMERIC DATA
                 .
                 .
                 .
DS2        DATASET                                                 X
                 DD1=ST2CDBC            DLBL FILE NAME - SECOND DATASET
           SEGM                                                    X
                 NAME=STPCORD,           SEGMENT NAME CUSTOMER ORDER   X
                 PARENT=STSCLOC,         PARENT IS CUST. LOCATION SEGM X
                 BYTES=55,               DATA LENGTH                   X
                 POINTER=TWINBWD,        BOTH PHYS.TWIN FWD AND BWD    X
                 RULES=(PPV)             LOGICAL RELATIONSHIP RULES
```

Figure 2. DBD Definition with Multiple DATASET Statements (Part 1 of 2)

```
                FIELD                                                   X
                    NAME=(STQCODN,SEQ,U),    UNIQUE KEY FIELD (DATE & ORD #) X
                    BYTES=12,                FIELD LENGTH                   X
                    START=1,                 WHERE IT STARTS IN SEGMENT     X
                    TYPE=C                   ALPHAMERIC DATA
                    .
                    .
                    .
    DS3     DATASET                                                      X
                DD1=ST3CDBC              DLBL FILE NAME - THIRD DATASET      SEGM
                    NAME=STCCITM,            SEGMENT NAME LINE ITEM         X
                    PARENT=((STPCORD,        PHYSICAL PARENT IS CUSTOMER ORD X
                    SNGL),                   PHYS. CHILD FIRST PTR. ONLY    X
                    (STPIITM,                LOGICAL PARENT IS ITEM INFORMAT X
                    V,                       REQUIRED FOR IMS/VS UPWARD COMP X
                    STDIDBP)),               LOG.PARENT IS IN ITEM DATA BASE X
                    BYTES=38,                DATA LENGTH                    X
                    POINTER=(TWINBWD,        BOTH PHYS. TWIN FWD AND BWD    X
                    LTWINBWD),               BOTH LOG. TWIN FWD AND BWD     X
                    RULES=(PVV)              LOGICAL RELATIONSHIP RULES
                FIELD NAME=STKCIIN,          INVENTORY ITEM NUMBER          X
                    BYTES=6,                 FIELD LENGTH                   X
                    TYPE=C                   ALPHAMERIC DATA
    *
    DS4     DATASET                                                      X
                DD1=ST4CDBC              DLBL FILE NAME - FOURTH DATASET     SEGM
                    NAME=STSCSTA,            SEGMENT NAME CREDIT STATUS     X
                    PARENT=STSCCST,          PARENT IS CUST. NAME/ADDR SEGM X
                    BYTES=24,                DATA LENGTH                    X
                    POINTER=TWIN,            PHYSICAL TWIN FWD ONLY         X
                    RULES=(,FIRST)           INSERT NEW SEGMS FIRST NOT LAST
    *                                        NOTE THERE IS NO KEY FIELD
    *                                        DESCRIBED FOR THE ABOVE SEGM
                SEGM                                                    X
                    NAME=STSCHIS,            SEGMENT NAME CUSTOMER HISTORY  X
                    PARENT=STSCCST,          PARENT IS CUST. NAME/ADDR SEGM X
                    BYTES=(130,53),          SEGMENT IS VARIABLE LENGTH     X
                    COMPRTN=DLZSAMCP,        NAME OF COMPRESSION ROUTINE    X
                    POINTER=TWINBWD          BOTH PHYS. TWIN FWD AND BWD
                FIELD                                                   X
                    NAME=(STQCHDN,SEQ,U),    UNIQUE KEY FIELD (DATE & ORD #) X
                    BYTES=12,                FIELD LENGTH                   X
                    START=3,                 WHERE IT STARTS IN SEGMENT     X
                    TYPE=C                   ALPHAMERIC DATA
                DBDGEN                   REQUIRED TO MARK DBD END
                FINISH                   FOR SOURCE COMPAT WITH IMS/VS
```

Figure 2. DBD Definition with Multiple DATASET Statements (Part 2 of 2)

Support of CICS Transaction Server Storage Protection

When operating the CICS Transaction Server-DL/I online system with storage protection (STGPROT=YES in SIT), DL/I allocates its system resources in storage protected with a partition key. For example:

- Online nucleus DLZNUCxx (loaded by CICS Transaction Server)
- DBD and PSB control blocks
- Buffer pool
- Access modules

This prevents them from being inadvertently overwritten through online programs, provided the programs are running in user key. To run online programs with user key, you have to define in the CDS file:

- Transactions with **TASKDATAKEY=USER**
- Programs with **EXECKEY=USER**

More information on the CICS Transaction Server storage protection can be found in the related CICS documentation.

Conditional SVA Loading

DL/I DOS/VS 1.10 and DL/I VSE 1.11 use different levels of phases, which are not compatible. Therefore the use of DL/I phases from the SVA may be critical when running in a CICS coexistence environment.

For DL/I VSE 1.11 a new parameter is available in the DLZACT TYPE=CONFIG macro. It allows control of the loading of SVA-eligible DL/I phases during CICS/DLI online initialization:

SVA={NO|YES}.

The default is NO.

When SVA=NO is specified, all DL/I phases will be loaded into partition space and used from there, even if they originally reside in the SVA.

When SVA=YES is specified, DL/I will use the phases from the SVA, if they reside there. Otherwise it will load them into partition space and use from there.

For further details, refer to "CICS Coexistence Environment" on page 18.

Enhancements Introduced via APARs

The following enhancements are available for DL/I VSE 1.11 as APARs. The corresponding APAR numbers are shown in parenthesis.

Return UIB Address in User DIB at EXEC DLI (PQ37331)

```
┌──────────────── Programming Interface Information ────────────────┐
```

In the HLPI (EXEC DLI) programming environment, DL/I status information is returned in the variables of the DL/I Interface Block (User DIB). This is different from the DL/I call-level programming environment, where DL/I status information is primarily returned through the PCB control block.

When a CICS/DLI online program implemented with the HLPI interface has scheduled a PSB and invokes another program implemented with the call-level interface that should use the same PSB and the environment is CICS/VSE 2.3 with DL/I DOS/VS 1.10, the call-level program could use the field TCADLPCB from the CICS TCA to retrieve the addresses of the PCBs.

As with the CICS Transaction Server access to the TCA control block is no more possible for application programs, the new field **DIBUIBA**, has been introduced in the User DIB. This allows the call-level program to obtain the PCB addresses in the following way:

1. The HLPI program schedules the PSB and retrieves the address of the UIB control block through field DIBUIBA.
2. The HLPI program invokes the call-level program passing the address of the UIB to it.
3. The call-level program obtains the addresses of the PCBs through field UIBPCBAL.

When the call-level program has retrieved the PCB addresses, it can enter DL/I calls to the same PCBs as the HLPI program, which had originally scheduled the PSB.

```
└──────────────── End of Programming Interface Information ────────────────┘
```

Allow variable for PSB name in DL/I HLPI Scheduling call (PQ39433)

```
┌──────────────── Programming Interface Information ────────────────┐
```

The HLPI (EXEC DLI) syntax of the DL/I Scheduling call has been extended that in addition to constants, variables are also supported to specify the name of the PSB to be scheduled. When a variable is used, it has to be enclosed in double parenthesis as shown below:

```
COBOL:    EXEC DLI SCHEDULE PSB((psb_name)) END-EXEC.

PL/I:     EXEC DLI SCHEDULE PSB((psb_name));
```

psb_name is the variable containing the name of the PSB to be scheduled.

```
└──────────────── End of Programming Interface Information ────────────────┘
```

DL/I AIBTDLI Interface (PQ39683)

Programming Interface Information

The AIBTDLI interface allows programs running in a VSE batch partition to issue DL/I calls without a DL/I batch environment having been established using DLZRRC00 or DLZMPI00.

A detailed description of this interface can be found in the *z/VSE e-business Connectors User's Guide*, SC33-8231.

End of Programming Interface Information

More than 25 Characters in PARM-String for DLZCTRL (PQ54003)

The 'PARM=' string passed to the DL/I dynamic initialization program DLZCTRL via the EXEC statement, can now also have more than 25 characters, if the DL/I specific part of it - the first 25 characters - corresponds to the syntax rules as documented in "Multiple MPS Systems" on page 1. DLZCTRL does not check the text following the 25th character. This part is now available to be interpreted by the program that is executed.

Two examples of valid 'PARM=' strings are shown in the following figure:

```
PARM='PARTID=F2,APPLID=DBDCCICS'              (old format with 25 characters)
PARM='PARTID=F2,APPLID=        /anytext...'   (new extended format)
```

Figure 3. Examples of Valid 'PARM=' strings

DTIMOUT does not work for DL/I BBCALL waits (PQ65658)

On the CICS Transaction Server platform, DL/I waits are normally issued via the CICS Black Box (BBCALL) interface. These waits are not sensitive to the CICS DTIMOUT parameter and therefore CICS/DLI tasks will not be purged at a time-out condition while they are suspended in a BBCALL wait.

When the system runs under heavy load with many concurrent CICS/DLI tasks, the DL/I Program Isolation (PI) function may have to serialize DL/I database accesses in a way that many tasks are held in a (BBCALL) wait for a longer period of time and the CICS/DLI system appears like being in a hang.

In order to allow that PI-waits can be resolved automatically by purging DL/I tasks through the CICS time-out feature after a certain wait time, a new DL/I parameter has been introduced for the DLZACT TYPE=CONFIG macro:

```
PTIMOUT={YES|NO}
```

When PTIMOUT=YES is specified, the CICS DTIMOUT parameter will also be effective for DL/I PI-waits. All other types of DL/I (BBCALL) waits are not affected and will not be DTIMOUT-purged as before.

When PTIMOUT=NO is specified, the CICS DTIMOUT parameter will not be effective for DL/I (BBCALL) waits as before.

The default is NO.

This APAR requires that the corresponding APAR PQ65543 of the CICS Transaction Server has also been installed.

DL/I Support for LE/VSE UADUMP Option (PQ68093)

When the LE/VSE UADUMP option is set, LE/VSE may create a system dump of the user partition when an abnormal termination has occurred. At the same time, if it was a DL/I batch application, DL/I dump processing is controlled through the UPSI statement, which may independently create a partition dump as well.

APAR PQ68093 prevents that both LE/VSE and DL/I take a partition dump while allowing full LE/VSE and DL/I abend handling to take place. When UADUMP has been defined and it is a DL/I batch application, the user should also request DL/I abend handling and a partition dump through UPSI bits 5 and 7 (set to zero). If an abnormal termination occurs with these settings, either LE/VSE writes the partition dump and the DL/I abnormal termination routine suppresses the duplicate one, or, if LE/VSE was not able to take the dump, DL/I will do it instead.

Misleading Result Information in DLZMDLI0 for QURY Call (PQ81067)

The result information returned from the DL/I online test program DLZMDLI0 for the QURY-function has been changed. The RESULTS-field on the screen will now display

STARTED   when the database was started, or

STOPPED   when the database was stopped

In both cases, a blank UIB status will be shown.

Scheduling Error When PSB is Generated with LANG=PLI (PQ92038)

The restriction that PSBs of language-type PL/I (LANG=PLI parameter in PSBGEN statement) are not supported for programs using the AIBTDLI interface has been removed. In addition, there is no dependency between the language of the PSB and the language of the program when the AIBTDLI interface is used. For programs implemented in the normal DL/I HLPI or call-level interface, the same rules apply as before.

For a description of the AIBTDLI interface, refer to the manual *z/VSE e-business Connectors User's Guide*, SC33-8231.

GETVIS-24 PSB Copies Allocated in Separate Subpool (PK15772)

When the parameter PSBLOC=GV24 has been defined in the DL/I online nucleus generation (DLZACT TYPE=CONFIG macro), the copies of the PSBs being scheduled are allocated in the 24-bit area of the partition GETVIS storage. In situations of heavy workload or perhaps specific unexpected conditions, the CICS/DLI system may more and more use up this 24-bit GETVIS storage or even run out of it. In this case, it is helpful to determine how much of this space has been allocated by DL/I, so that appropriate actions can be taken.

For this reason, the DL/I GV24 PSB-copies will now be allocated in their own separate subpool with the name **DLIPSB**. This allows to tell, for example, via the attention GETVIS command, how much of the 24-bit GETVIS storage has been consumed by DL/I at a certain point of time.

DL/I DOS/VS 1.10 Enhancements

The following enhancements are available with DL/I VSE 1.11, but have also been made available via APARs for DL/I DOS/VS 1.10 previous to the shipment of DL/I VSE 1.11. The corresponding APAR numbers are shown in parenthesis.

Processing Options N and T for DL/I Batch Environment (PN30337)

The combination of processing option N or T with the processing option GO or GOP will also be effective in the DL/I batch environment.

When DL/I detects an invalid pointer during the access to a database segment (because of concurrent updates from a CICS/DLI online partition, for example), PROCOPT=N returns status code GG, and PROCOPT=T performs an automatic retry instead of issuing an error message and abnormally terminating the application.

A detailed description of the processing options N and T can be found in the DL/I manual *Resource Definition and Utilities*, page 154.

Change Accumulation with Selected Logfiles (PN43036)

The existing Change Accumulation process required that for every database for which DL/I log change records were passed to the Change Accumulation process, there had to be a corresponding DL/I database OPEN record present on the input logfile(s). If this was not the case, Change Accumulation assumed that some logfile might have been missing, issued message DLZ375I, and ended abnormally.

While this ensured that the set of input logfile(s) was complete, it also prevented the Change Accumulation from running with only a selected single logfile. For example, it was not possible to run Change Accumulation with the second (or higher) part of a large logfile only. This made prompt recovery actions difficult in some cases.

The Change Accumulation utility will now also accept logfiles that do not contain database OPEN records for all databases with log change records. When such a condition is detected the new message DLZ431I is issued instead of DLZ375I. This is a warning and should remind the user that some logfile(s) may be missing or an incorrect logfile has been mounted in the input stream.

DUMP Macro with RC=12 (PN47959)

In case of an abnormal termination, the DL/I modules DLZOLI00, DLZURDB0, DLZUDMP0, DLZTPRT0, DLZLOGP0, DLZURGL0 and DLZURRL0 will now call the VSE DUMP macro with RC=12 instead of RC=0. This allows the usage of the ON $RC JCL statement to prevent the execution of subsequent job steps, if one of these modules has failed.

No Unload of Tape Work Files - DLZURG10/DLZURGP0 (PN56301)

When running the DL/I Prefix Resolution utility DLZURG10 and the DL/I Prefix Update utility DLZURGP0 a new control statement may be used to invoke different rewind options, when some of the utility workfiles have been assigned to tape, instead of unconditionally unloading the tape. This allows the submission of

the complete Reorganization utility job sequence (Pre-reorg, Unload, Reload, Prefix-resolution and Prefix-update) repetitively without operator intervention.

The usage of the new control statement is optional. When specified, it is active for the following work files, when they are assigned to tape:

```
-  DLZURG10:   WRKINxx   SYS013
               INTRMED   SYS010
-  DLZURGP0:   WORKFIL   SYS011
               INDXWRK   SYS014
```

The format is as follows:

```
  REW=option
```

Columns 1-4 must contain REW= and column 5 must contain one of these letters to indicate the desired tape option:

```
  N - means no rewind
  R - means rewind only, and
  U - means rewind and unload.
```

No other columns are used. When the new control statement is not entered, tapes will be unloaded by default during termination of the utility as before.

Support for DL/I 31-Bit Applications (PN67649)

DL/I applications can run in 31-bit mode and above the 16 MB line without restriction. Compilers, which may be used to create DL/I 31-bit applications are:
* PL/I for VSE/ESA
* COBOL for VSE/ESA
* COBOL II
* High Level Assembler

31-bit execution is possible both for DL/I online and batch/MPS applications, implemented with the DL/I CALL or HLPI interface. DL/I call parameters can reside above the 16 MB line.

User-written DL/I exit routines (such as randomizing and compression routines) can not run in 31-bit mode.

Appendix A, "Sample Job Streams for DL/I 31-Bit Applications," on page 77 provides sample job streams which show how to create DL/I 31-bit applications in different environments.

Additional Support for LE/VSE

The following LE/VSE functions interacting with DL/I are supported:
* TRAP runtime option
* CEETDLI interface

LE/VSE TRAP Runtime Option

DL/I batch applications implemented in one of the LE/VSE-conforming languages can run with the LE/VSE TRAP runtime option being active. When TRAP(ON) is set, any abnormal termination condition caused by the application will first be processed by LE/VSE error routines. If DL/I is to receive control again after LE/VSE error handling, for example, to write back pending DL/I data buffers and correctly close the DL/I databases or log files:

```
ABTERMENC(ABEND)
```

or any equivalent runtime option must have been specified. Only this ensures proper DL/I termination, when TRAP(ON) has been set.

When running with TRAP(ON), DL/I STXIT linkage is overlaid by LE/VSE STXITs. Any DL/I dumps issued by one of the DL/I exits after message DLZ001I will not reflect the original storage contents from the time, when the error had occurred. These DL/I dumps can be suppressed by the according UPSI byte setting as described in the *DL/I manual Resource Definition and Utilities* (SH24-5021-02).

When original DL/I STXIT linkage and exit processing is preferred, the application should be run with TRAP(OFF). Information on DL/I STXIT linkage can be found in the DL/I manuals *Resource Definition and Utilities* referenced above and *Guide for new Users* (SH24-5001-4).

For information on the LE/VSE TRAP option, refer to the corresponding LE/VSE documentation.

LE/VSE CEETDLI Interface

―――――――――――――― **Programming Interface Information** ――――――――――――――

LE/VSE provides a callable service, CEETDLI, that can be used to issue DL/I calls. CEETDLI in general performs the same functions as the existing language-specific interfaces (ASMTDLI, CBLTDLI, PLITDLI), but is language independent. Only LE/VSE conforming applications can call CEETDLI. Calls to CEETDLI are implemented in the same way as calls to the language-specific interfaces. A description of the CEETDLI syntax can be found in the LE/VSE Programming Guide.

The existing language-specific interfaces will continue to be supported both in existing programs and programs newly compiled and linked in the new LE/VSE environment.

――――――――――― **End of Programming Interface Information** ―――――――――――

Increased VSAM Blocksize for DL/I Index Databases (PN68583)

The maximum blocksize for DL/I index databases (VSAM KSDS) has been increased to 30720 bytes. During DBD-generation DL/I calculates the blocksize from the values entered at the BLOCK and RECORD parameters of the DATASET statement. New possible blocksizes are:
- 512 bytes - 8192 bytes in steps of 512 bytes
- 8192 bytes - 30720 bytes in steps of 2048 bytes

The maximum segment size has not been changed.

Command Code U for HLPI Programs (PN73378)

―――――――――――――― **Programming Interface Information** ――――――――――――――

The command code U will also be supported in the DL/I HLPI language through the new option KEEP.

KEEP can be used with all types of DL/I GET calls and the DL/I ISRT call. It requires the same position as the existing options FIRST or LAST. It should not be

specified on the same segment level together with FIRST or LAST. Coding example:

```
        EXEC DLI GET NEXT IN PARENT USING PCB(2)
                   FIRST SEGMENT (SEGMX)
                   KEEP  SEGMENT (SEGMY)
                         SEGMENT (SEGMZ) INTO (IOAREA)
```

A general description of the command code U can be found on page 4-3 of the *DL/I CALL and RQDLI Interfaces* manual (SH12-5411-6). The same rules and restrictions mentioned there also apply to the HLPI-style usage. The DL/I HLPI language is documented in the *DL/I HLPI* manual (SH24-5009-2).

─────── **End of Programming Interface Information** ───────

New System Call QURY (PN85936)

─────── **Programming Interface Information** ───────

The new DL/I system call QURY is available for DL/I online applications. It returns the status of a DL/I database.

The DL/I sample program for system calls, DLZMDLI0, allows the selection of the new QURY call.

─────── **End of Programming Interface Information** ───────

Year 2000 Support (PN87288)

DL/I has to process date values when they are passed as part of:

- log records
- utility records (like file header or detail record)
- utility control statements.

To support Year 2000, the existing 2-digit year values of the mentioned DL/I records and control statements will be interpreted according to a fixed-window scheme:

```
     yy = 50->99 = 19yy
     yy = 00->49 = 20yy
```

The date format of the physical records and control statements remains unchanged. All date values printed by DL/I utilities (such as page headers, log print records, certain start messages) will show the year in a 4-digit format.

Access to a DL/I database is not affected by the Year 2000 conversion.

GETVIS for Program Isolation Storage Above 16 MB (PN88972)

Depending on the DL/I application design (e.g type of data base calls used and TERM/CHKP call frequence) the program isolation function may use up to several megabytes of working storage. DL/I will now always attempt to obtain this working storage from the GETVIS-31 area. When not enough GETVIS-31 storage is available, it will be requested from the GETVIS-24 area as before.

More Than 32 HD-Buffers per Subpool (PN89468)

The maximum number of HD-buffers for VSAM ESDS datasets, which can be allocated in a subpool has been increased to 255. The definition of the desired number of HD-buffers is done as before:

- In the DL/I batch environment via the HDBFR statement in the DLI parameter statement.

```
       HDBFR=(nn,dbdname1,dbdname2,..)
```

- In the DL/I online environment via the DLZACT TYPE=BUFFER macro in the ACT generation job.

```
       DLZACT TYPE=BUFFER,HDBFR=(nn,dbdname1,dbdname2,...)
```

The number of buffers nn can be specified in a range from 2 to 255. The default value of 32 is unchanged.

PSBs Above 16 MB (PQ09904)

During CICS/DLI initialization, PSBs may be loaded into GETVIS-31 storage of a CICS/DLI online partition. The location of the PSBs is controlled through the new PSBLOC parameter of the DLZACT TYPE=CONFIG macro in the ACT generation job:

```
       PSBLOC={ANY|BELOW}
```

When ANY is specified, DL/I will try to load the PSBs into GETVIS-31 storage. If there is not enough GETVIS-31 storage available, the PSBs will be loaded into the GETVIS-24 area.

When BELOW is specified, the PSBs will be loaded into the CICS DSA as before. This is the default.

When the PSBs are residing above the 16MB line, every CICS/DLI task scheduling a PSB will get its own copy in the DSA. That means the original PSB is copied over from GETVIS-31 to the DSA and processed there. This is necessary, because the DL/I action modules are executing in amode-24. At DL/I task termination the storage of the PSB copy is returned to the DSA.

In this way, the whole set of PSBs defined in the DL/I online nucleus and initially loaded during CICS/DLI startup, can be held in GETVIS-31 storage. DSA storage for PSBs will only be needed on demand from a DL/I task.

Migrating to DL/I VSE 1.11 and the CICS Transaction Server for VSE/ESA 1.1

DL/I VSE1.11 is upward compatible from DL/I DOS/VS 1.10. Programming interfaces have not been changed. DL/I batch applications and online applications implemented in CICS command level language will run as before. CICS online applications implemented in CICS macro style or accessing CICS control blocks directly need to be converted as shown in "CICS Transaction Server for VSE/ESA 1.1 Migration Items" on page 17.

Migration aspects related to the enhancements of DL/I VSE 1.11 are described below. If you are migrating from an earlier DL/I release than DL/I DOS/VS 1.10, you should also refer to Chapter 4, "Migration and Compatibility Considerations," on page 27.

For specific migration aspects concerning the CICS Transaction Server, refer to "CICS Transaction Server for VSE/ESA 1.1 Migration Items" on page 17.

DL/I VSE 1.11 Migration Items

Changed Modules of DL/I Online Nucleus
All DL/I online users have to reassemble the online nucleus with the DL/I DLZACT macro.

Storage Layout Control Table
The use of the DL/I Storage Layout Control (SLC) table is no longer supported.

Multiple MPS Systems
The new PARTID/APPLID parameters must be specified in an MPS batch job, when multiple CICS/DLI MPS systems are active. This also applies, when invoking MPS via the MPS/batch selection program DLZCTRL.

More than 4 GB for HD Databases
When a HD database is to be reallocated on multiple VSAM datasets, its DBD and PSBs have to be rebuilt and the database reorganized as described in "More than 4 GB for HD Databases" on page 3.

Processing Option N for DL/I Batch Environment
PROCOPT=N returns status code GG instead of abnormal termination if invalid DL/I pointers are detected. DL/I batch applications may have to be changed to examine the new status code.

Change Accumulation with Selected Logfiles
The completeness check for input logfiles of the Change Accumulation Utility has been changed. Refer to "Change Accumulation with Selected Logfiles (PN43036)" on page 11 for details.

Dump Macro with RC=12
Some DL/I programs as described in "DUMP Macro with RC=12 (PN47959)" on page 11 are now terminating with RC=12 instead of RC=0, when an abend situation has been detected. Jobs using conditional JCL may have to be adapted accordingly.

Support for DL/I 31-Bit Applications
DL/I applications, which are to exploit the 31-bit support of the new LE/VSE compilers or the High Level Assembler for VSE, have to be re-compiled and re-linked as described in Appendix A, "Sample Job Streams for DL/I 31-Bit Applications," on page 77.

Existing applications (24 or 31 bit addressing) run as before.

You have to specify PATH sensitivity for all segments, for which you do path retrieve, insert or replace calls from 31-bit applications. This applies both for CALLDLI and EXEC DLI statements. A new PSBGEN and ACBGEN has to be run

for all affected PSBs. For changes required for the new CICS Transaction Server online environment, refer to "CICS Transaction Server for VSE/ESA 1.1 Migration Items."

LE/VSE TRAP Runtime Option

The abend handling for DL/I batch and MPS batch programs works slightly differently, when running with the LE/VSE runtime option TRAP(ON). Refer to the description of the "LE/VSE TRAP Runtime Option" on page 12.

Increased VSAM Blocksize for DL/I Index Databases

When the DATASET statement of an index DBD has been changed to define a different blocksize, a new DBDGEN has to be done for this index database. Then ACBGEN must be rerun for any PSBs, which may access the database.

PSBs above 16 MB

The new PSBLOC parameter becomes active by reassembling the online nucleus with the DL/I DLZACT macro.

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.

Users operating VSE 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.

CICS Transaction Server for VSE/ESA 1.1 Migration Items

Applications

CICS/DLI macro level programs have to be converted to the CICS command level.

Adaptions are also required for CICS/DLI online programs which have been written using the DL/I CALLDLI interface, and which are using the CICS TCA to check return codes (TCAFCTR) or retrieve the PCB address list (TCADLPCB). This is needed because the CICS TCA can no longer be accessed. The equivalent fields can be retrieved from the DL/I UIB control block instead.

Chapter 3 of the *DL/I CALL and RQDLI* manual (SH12-5411-6) contains examples showing the syntax for DL/I calls, when the UIB is used.

CICS Transaction Server - DL/I Tables Requirements

The CICS components needed for DL/I online execution have to be defined according to the description in the *DL/I Resource Definition and Utilities* manual (SH24-5021-02), pages 255-263.

The CICS PCT and PPT tables no longer exist. These entries have to be specified using the CICS RDO facility.

All DL/I user and system programs should be defined with EXECKEY(USER) and the associated transactions with TASKDATAKEY(USER). The MPS master partition controller DLZMPC00 no longer needs a definition for a CICS transaction workarea (TWA) in the CSDB transaction entry.

Specify DBP=2$ in the CICS SIT to allow for dynamic transaction backout for DL/I programs.

CICS Coexistence Environment

As mentioned in the introduction of this chapter, starting with VSE/ESA 2.4, the CICS Transaction Server 1.1 and CICS/VSE 2.3 can coexist and run together with DL/I VSE 1.11 (CICS Transaction Server) and DL/I DOS/VS 1.10 (CICS/VSE).

In such an environment you can make use of the following functions in addition to regular CICS/DLI online transaction processing:

- CICS/DLI MRO function shipping for usage of remote or extended remote PSBs.
- Execution of MPS batch jobs. The release level of the DLZMPI00 root phase must correspond to the release level of the MPS master partition controller DLZMPC00 and the CICS/DLI online system it tries to connect to. MPS batch jobs from different releases may run at the same time.

DL/I DOS/VS 1.10 and DL/I VSE 1.11 are installed in separate sublibraries. Therefore, the correct LIBDEF chain is essential for loading the correct version of the DL/I modules.

In a coexistence environment, you should load SVA-eligible phases from the DL/I DOS/VS 1.10 sublibrary, ensuring that the DLZACT for DL/I VSE 1.11 has the parameter **SVA=NO** set. For further information about the SVA parameter, refer to "Conditional SVA Loading" on page 7.

New and Changed Messages

Refer to Chapter 9, "New and Changed Messages," on page 67 for details.

Chapter 2. Operating Environment

This chapter describes the machine and programming requirements as well as the programming environment for the operation of DL/I VSE 1.11.

Machine Requirements

Virtual Storage

The minimum main storage requirements for DL/I VSE 1.11 are the same as those needed for conventional operation of your VSE system. 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"

Disk Storage

The approximate library space (in library blocks) needed for a DL/I VSE 1.11 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 space to the values given below for adding new code and applying service.

```
Production Sublibrary:   6.200 library blocks
Generation Sublibrary: 12.700 library blocks
```

To find out the actual amount of library space required for the installation of DL/I VSE 1.11, scan the distribution tape using the VSE-provided installation dialog.

Tape Drive

You need at least one tape drive for the installation of DL/I VSE 1.11.

Using the IBM 3480 or 3490 as Logging Device

When you use the IBM 3480 or IBM 3490 tape devices 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, the CICS Transaction Server 1.1 supports the IBM 3480 and IBM 3490 tape devices. However, these units are *not* recommended for journaling.

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.

Running DL/I in a VM Environment

If you plan to run VSE under VM, you have to consider VM storage requirements.

For using DL/I with CICS in a VM virtual IMF and IUG Functions 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.

- 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.

Programming Requirements

The following IBM programs are required for the operation of DL/I VSE 1.11:

- **VSE/ESA 2.4 (5690-VSE)** (or later)
- **DFSORT/VSE 3.4 (5746-SM3)** (or equivalent)

Programming Environment

DL/I VSE 1.11 is designed to work with the following IBM programs:

- **VSE/ESA 2.4 Base Programs**
  - CICS Transaction Server for VSE/ESA 1.1
  - VSE/VSAM Space Management for SAM
  - VSE/ICCF
    DL/I batch and MPS batch jobs can run in VSE/ICCF interactive partitions.
- **VSE/ESA 2.4 optional program**
  - DB2 Server for VSE & VM 6.1 (or later)
- **VM/SP 3.1 or later**
  - VM (including VM/XA, VM/ESA and z/VM) supports the DL/I IMF and IUG functions through ISPF.

Compiler and Assembler Support

DL/I VSE 1.11 supports the following compilers and assemblers:

- **High Level Assembler for VSE 1.3 (or later)**
- **LE/VSE 1.4**
- **COBOL for VSE/ESA 1.1**
- **PL/I for VSE/ESA 1.1**
- **DOS/VS RPG II 1.3** (batch applications only)

Other Related Software Products

- **Data Management System/CICS (5746-XC4)**
  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 (5796-PKF)**
  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.

Operating Environment

Chapter 3. Installation

VSE provides dialogs to install optional programs. To install DL/I, use the dialog *Install Program(s) from Tape*. You reach this dialog via the *Install Programs-V2 Format* dialog. The dialogs are described in detail in the manuals *VSE/ESA Installation* or *z/VSE Installation*.

> **Note**
>
> Starting with VSE/ESA 2.4 you can install both, DL/I VSE 1.11 and DL/I DOS/VS 1.10 and run them together in a CICS coexistence environment (CICS Transaction Server 1.1 with DL/I VSE 1.11 and CICS/VSE 2.3 with DL/I DOS/VS 1.10). When installing a second DL/I on your system, you get the following message:
>
> ```
> M231D INSTALLATION WILL OVERWRITE PRODUCT ... ENTER "DELETE" OR "KEEP"
> ```
>
> Respond with KEEP if you want to run both versions of DL/I.

DL/I Shipment Details

DL/I VSE 1.11 is shipped as VSE optional program.

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

Note: DL/I IMF and IUG is shipped in its VM version only. Refer also to "DL/I IMF and IUG Functions" on page 33.

DL/I is partially delivered in object code and is shipped with a production and a generation sublibrary.

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
  - Sample programs of type A, C, P, and R (see "Source Books in the Production Sublibrary," below).

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 (supported in CICS/VSE only)

DL/I Database Definition and Load Samples:

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 (supported in CICS/VSE only)

DL/I High Level Assembler Sample Programs:

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

DL/I COBOL Sample Programs:

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 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 (supported in CICS/VSE only)

DL/I Documentation Aid, ISQL EXTRACT DEFINEs:

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

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).

Installation

Chapter 4. Migration and Compatibility Considerations

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

For the latest migration information, refer to "Migrating to DL/I VSE 1.11 and the CICS Transaction Server for VSE/ESA 1.1" on page 15.

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

- **1.7.0**

  Refer to "Migration from DL/I 1.7.0 to 1.7.1."

- **1.7.1**

  Refer to "Migration from DL/I 1.7.1 to 1.8.0" on page 28.

- **1.8.0**

  Refer to "Migration from DL/I 1.8.0 to 1.9.0" on page 28.

- **1.9.0**

  Refer to "Migration from DL/I 1.9.0 to 1.10.0" on page 28.

- **1.10.0**

  Refer to "Migrating to DL/I VSE 1.11 and the CICS Transaction Server for VSE/ESA 1.1" on page 15.

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

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" on page 28.

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."

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."

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 procedures as described below.

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" on page 62.

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" on page 63.

Migration from DL/I 1.10.0 to 1.11.0

Refer to "Migrating to DL/I VSE 1.11 and the CICS Transaction Server for VSE/ESA 1.1" on page 15

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 VSE
  - 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 VSE and HD RELOAD in IMS/VS
  - HISAM UNLOAD in DL/I VSE 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".

Chapter 5. Customizing DL/I

SVA Loading for DL/I

> **Note**
>
> When running DL/I VSE 1.11 (CICS Transaction Server) and DL/I DOS/VS 1.10 (CICS/VSE) together in a coexistence environment, the loading of DL/I phases into the SVA should be carefully planned. The phases are not compatible between the two DL/I releases. Refer also to "Conditional SVA Loading" on page 7 and "CICS Coexistence Environment" on page 18.

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 93K 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"
- "Loading User-Specified SVA Phases" on page 32
- "Loading SVA Phases via IPL" on page 32

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 4.

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

Figure 4. Sample Job SVALOAD1.
Use this job to load all SVA-eligible DL/I phases into the SVA.

Loading User-Specified SVA Phases

To load *selected* SVA-eligible phases, run a job similar to that shown in Figure 5.

```
// 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 5. Sample Job SVALOAD2.
Use this job to load selected SVA-eligible DL/I phases into the SVA.

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 4 on page 31
- Figure 5

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

CICS Adaptation

If you plan to:
- run DL/I online, or
- employ MPS,

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

1. Define DL/I databases and application programs:

   a. Define DL/I databases 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 CSD file. Specify TASKDATAKEY(USER) and EXECKEY(USER).

2. Create DL/I and CICS system tables 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.

   With DL/I VSE 1.11, support of the Storage Layout Control (SLC) table has been dropped.

b. In a CICS XRF environment, if DL/I program isolation is active, or CICS emergency restart or dynamic transaction backout is to be used with DL/I tasks, assign the DL/I log to the CICS system log. In this case, specify DBP=2$ in the CICS System Initialization Table (SIT). The DL/I log is assigned by use of the VSE UPSI byte information. Refer to the following manual for information on how to use this UPSI byte with CICS:

- *DL/I Guide for New Users*, under the "Chapter: Online and MPS Considerations, UPSI Byte Settings (Online)"

c. 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 add the DL/I run and buffer statistics function to your CICS system by defining:

- DLZSTTL as program, and
- CSDE as transaction in the CSD file.

d. 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 application program (`DFHTDLI`) is supplied with the pre-generated CICS system. For the description of the program, refer to the following manual:

- *CICS Transaction Server* 1.1 System Definition Guide

For a full description of all DL/I related CICS tables, refer to the manual *DL/I Resource Definition and Utilities*, pages 255-263.

For a description of how to use DL/I and CICS XRF, refer to "Features Introduced with DL/I DOS/VS 1.9" on page 57.

DL/I IMF and IUG Functions

> **Note**
>
> DL/I IMF and IUG needs ISPF as a prerequisite.
>
> While ISPF is not supported by VSE, you can use DL/I IMF and IUG functions when ISPF is running under VM/CMS.
>
> The VSE version of DL/I IMF and IUG is no longer delivered with DL/I VSE1.11.

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".

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.

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*.

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 6 on page 35.

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

Figure 6. Sample Job DELETVM

DL/I Documentation Aid and the ISQL EXTRACT DEFINEs Utility

> **Note**
>
> The following information on the DL/I Documentation Aid facility refers to SQL/DS. It also applies to the VSE optional program DB2 for VSE & VM to the extent that DB2 is upward compatible to SQL/DS.
>
> For the ISQL EXTRACT facility, you need SQL/DS prior to Version 3.

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.

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 on page 36). This job punches three DL/I Documentation Aid job streams (provided in the production sublibrary) using the VSE librarian. Provide the appropriate 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. Preprocess and reassemble the modules DLZDLBD and DLZDLBP with the SQL/DS release installed on your system.

   To do this, create and run a job similar to that shown in Figure 8 on page 38. This job stream creates the type A, type OBJ, and type PHASE entries for the modules.

   Figure 8 on page 38 shows processing for module DLZDLBD. For the module DLZDLBP, change the names according to Table 1 on page 37.

   Note: If during assembly of DLZDLBDP and DLZDLBPP, message ASMA044E "Undefined symbol - SQLDSIZ" occurs, remove the END statement in the last line of DLZDLBDP and DLZDLBPP, or put it at the source end.

   After this, or in general, whenever DLZDLBDP or DLZDLBPP have been reassembled, the following other modules of the DL/I Block Builder Utility DLZUACB0 have also to be reassembled and their objects catalogued in the DL/I production sublibrary:
   - DLZDLBL0
   - DLZDLBL1
   - DLZDLBL2
   - DLZDLBL3

   Then the DL/I Block Builder utility DLZUACB0 has to be re-linkedited from the new objects.

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 8 on page 38.

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

ISQL EXTRACT DEFINEs

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

You must first preprocess and reassemble the module DLZEXDF with the SQL/DS release installed on your system. This has to be done according to the naming convention shown in Table 1 and using the sample job shown in Figure 8 on page 38.

Note: If during assembly of DLZEXDFP, message ASMA044E "Undefined symbol - SQLDSIZ" occurs, remove the END statement in the last line of DLZEXDFP, or put it at the source end.

Table 1. Naming Convention for DL/I DA and ISQL EXTRACT Modules

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

Note that in Figure 8 on page 38:
- 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

Customizing DL/I

```
// 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
* ---------------------------------------------------------- *
```

Figure 8. Sample Job PREP DLZDLBD (Part 1 of 2)

```
* 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 8. Sample Job PREP DLZDLBD (Part 2 of 2)

Chapter 6. Post Installation Tasks and Tools

Getting DL/I Sample Application Job Streams

To verify the correct installation of DL/I VSE 1.11, 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" on page 23).

Creating an MSHP Retrace

After you have installed DL/I VSE 1.11, you may wish to print the DL/I related information contained in the System History File.

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

   4 (Retrace History File), and then

   7 (Retrace Component ID)

2. Run a job as shown below:

```
// JOB RETRACE
// EXEC MSHP,SIZE=900K
    RETRACE COMP ID=5746-XX1-00-1IO
/*
/&
```

Displaying DL/I Library Directory Listings

To display the directory listings of your DL/I VSE 1.11 sublibraries, run the jobs in Figure 9 and Figure 10:

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

Figure 9. 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 10. 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.

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 DL/I Components" on page 45.

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.

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.

Debugging Tools

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".

Chapter 7. Applying Service

Service to DL/I modules is usually provided in the form of a PTF (Program Temporary Fix). To apply a PTF, use the following dialogue path starting from the main menu of the VSE Interactive Interface:

1   Installation

4   IBM Service

2   PTF Handling

Sometimes you can also apply service by means of an APAR (Authorized Program Analysis Report) fix. In this case you have to run the update job supplied with the APAR documentation or distributed through the IBM support center. The update job re-assembles and re-links the module(s) to be serviced.

Error Messages while Reassembling 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. They can be ignored and the assembly has completed successfully. Described below are the assembler messages you may encounter:

- ASMA033I Storage alignment for *label* unfavorable

  This warning message may be issued during the assembly of the DL/I modules:

  ```
  DLZTPRT0 and DLZTRPR0.
  ```

- ASMA300W USING overridden by a prior active USING on statement number *nn*.

  This warning message may be issued during the assembly of the following DL/I modules:

  ```
  DLZBACM0, DLZCUMM0, DLZDHDS0, DLZDLBL0, DLZDLBL1, DLZDLOC0, DLZDLRA0,
  DLZDLRB0, DLZDLRC0, DLZDLRD0, DLZDLRE0, DLZDLRF0, DLZDMPM0, DLZDXMT0,
  DLZLGPM0, DLZRDBC0, DLZRDBM0, DLZRRC00, DLZTXIT0.
  ```

- ASMA301W Prior active USING on statement number *nn* overridden by this USING

  This warning message may be issued during the assembly of the following DL/I modules:

  ```
  DLZDDLE0, DLZDLA01, DLZDLBL0, DLZDLBL1, DLZDLBL2, DLZDLBL3, DLZDLD00,
  DLZDLOC0, DLZDLTXX, DLZDSEH0, DLZDXMT0, DLZFSDP0, DLZFTDP0, DLZLOC00,
  DLZLOGP0, DLZMPI00, DLZODP,  DLZUDMP0, DLZURDB0, DLZURGL0, DLZURGS0,
  DLZURGU0, DLZURG10.
  ```

- ASMA302W USING specifies register 0 with a non-zero absolute or relocatable base address

  This warning message may be issued during the assembly of the following DL/I modules:

  ```
  DLZDLRA0, DLZDLRB0, DLZDLRC0, DLZDLRD0, DLZDLRE0, DLZDLRF0, DLZDLRG0.
  ```

- ASMA303W Multiple address resolutions may result from this USING and the USING on statement number *nn*

  This warning message may be issued during the assembly of the following DL/I modules:

```
DLZBACK0, DLZBNUC0, DLZCBDP0, DLZDBH00, DLZDHDS0, DLZDLA00, DLZDLBDP,
DLZDLBL1, DLZDLBL2, DLZDLBL3, DLZDLBPP, DLZDLD00, DLZDLRA0, DLZDLRB0,
DLZDLRC0, DLZDLRD0, DLZDLRE0, DLZDLRF0, DLZDLRG0, DLZDLTXX, DLZDPSB0,
DLZDSEH0, DLZDXMT0, DLZEIPB1, DLZEIPO0, DLZEXDFP, DLZLBLM0, DLZLOGP0,
DLZMPI00, DLZODP,   DLZOLI00, DLZPRPAR, DLZPRURC, DLZQUEF0, DLZRDBL0,
DLZRDBL1, DLZRDBM0, DLZRRC00, DLZUACB0, DLZUCUM0, DLZUDMP0, DLZURDB0,
DLZURGL0, DLZURGP0, DLZURGS0, DLZURGU0, DLZURG10, DLZURRL0, DLZURUL0,
DLZTPRT0.
```

Error Messages while Relinking DL/I Components

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. In most of the cases the link edit process continues successfully. 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`, `DLZNUCxx`, 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.

Applying Service

Chapter 8. Features Introduced with Earlier Releases of DL/I

Features Introduced with DL/I DOS/VS 1.7.1

The features 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.

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.

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 11 on page 50). 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 11, 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≠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 11. Sample Program DLZCTRL

The New GSTA Call

┌──────────────── **Programming Interface Information** ────────────────┐

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).

└──────────────── **End of Programming Interface Information** ────────────────┘

GSTA Call Format

┌──────────────── **Programming Interface Information** ────────────────┐

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 12 on page 52.

Earlier Releases of DL/I Features

```
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 12. How to Invoke the New GSTA Call in an Assembler Program

──────────── **End of Programming Interface Information** ────────────

GSTA Return Codes

Programming Interface Information

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

Table 2. GSTA Call Return Codes

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.

End of Programming Interface Information

Layout of the Statistics Buffer

With the 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 13. Layout of the Statistics Buffer DLZSTBF. **In the job statistics, the indicated numbers of DL/I calls include DL/I internal calls.**

Features Introduced with DL/I DOS/VS 1.8

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.

Literal String in the HLPI WHERE Clause

--- Programming Interface Information ---

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*.

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*.

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*.

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*.

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*.

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*.

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 9, "New and Changed Messages," on page 67.

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).

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.

End of Programming Interface Information

Features Introduced with DL/I DOS/VS 1.9

DL/I 1.9 supports the VSE/ESA dynamic partitions, an enhanced database I/O error handling and the CICS/VSE 2.1 XRF function.

Dynamic Partition Support

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*.

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 9, "New and Changed Messages," on page 67.

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.

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.

DL/I in a CICS/XRF Environment

VSE 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.

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).

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*.

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.

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
```

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.

Features Introduced with DL/I DOS/VS 1.10

The following enhancements are available with DL/I DOS/VS 1.10:
- DL/I Applications above the 16MB line of Storage
- VSE/VSAM HS-Buffers Above the 16MB Line of Storage
- Virtual Disk Exploitation
- DL/I Run Statistics (DLZSTTL) Enhancements

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.

DL/I Applications above the 16MB Line of Storage

Note: This support has been further enhanced. Refer to "Support for DL/I 31-Bit Applications (PN67649)" on page 12 for a complete description and the latest information.

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.

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" on page 33, and the manuals:
- *DL/I Guide for New Users*
- *DL/I Interactive Resource Definition and Utilities*

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."

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.

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*.

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 14 on page 65 shows an example of the output of the DL/I Run Statistics program DLZSTTL.

```
DLZSTTL ================================================================
DLZSTTL     DL/I  DATA BASE- AND SUBPOOL-STATISTICS   1999/094   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
```

Figure 14. Example: Output of the DL/I Run Statistics Program (DLZSTTL) (Part 1 of 2)

Earlier Releases of DL/I Features

```
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  NUMBER OF   NUMBER OF   NUMBER OF
DLZSTTL POOL     OF DMBS  BUFFERS     SIZE      BH REQUESTS  FROM POOL   READ REQUSTS WRITES ISSUD  NEW BLKS
DLZSTTL ----I----------I----------I---------I-------------I-------------I-------------I-------------I--------------
DLZSTTL  1        1         8        512          -            -            -            -            -
DLZSTTL  2        1        10       2048         1064          738          303          11           -
DLZSTTL  3        2        20       2048          39           32            6           -            -
DLZSTTL  4        2        32       4096          -            -            -            -            -
DLZSTTL
DLZSTTL ================================================================
DLZSTTL
DLZSTTL - INFORMATION PER DATA BASE -
DLZSTTL
DLZSTTL ----------(DL/I)---------I I------(VSAM)------I-----------(VSAM FOR HISAM,SHISAM AND INDEX ONLY)------------
DLZSTTL
DLZSTTL DBD-    DMB-    SUB- BUF- W  CI-     NUMBER OF   NUMBER OF   NUMBER OF   NUMBER OF   NUMBER OF   NUMBER OF
DLZSTTL NAME    ORGAN. POOL SIZE    SIZE      EXCPS     CI-SPLITS   CA-SPLITS   RETRIEVES    UPDATES     INSERTS
DLZSTTL -------I------I----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            -           -            1           -           1
DLZSTTL STDCX1P INDEX          2048        10            -           -           86           6           3
DLZSTTL -------I------I----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------------I-----------I------------
DLZSTTL TOTAL                              332           -           -           87           6           4
DLZSTTL -------I------I----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 14. Example: Output of the DL/I Run Statistics Program (DLZSTTL) (Part 2 of 2)

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=...
```

Chapter 9. New and Changed Messages

In this Manual ...

The following messages are new for DL/I VSE 1.11:
>
> DLZ1028
> DLZ1061
> DLZ1062
> DLZ019I
> DLZ093I
> DLZ094I
> DLZ143I
> DLZ144I
> DLZ145I
> DLZ146I
> DLZ147I
> DLZ250I
> DLZ251I
> DLZ431I

The following messages have been included earlier because they were new for DL/I DOS/VS 1.10:
>
> DLZ1059
> DLZ1060
> DLZ109I
> DLZ379I

The following messages have been included earlier because of changes for DL/I DOS/VS 1.10:
>
> DLZ1055
> DLZ002I
> DLZ015I
> DLZ037I
> DLZ058I
> DLZ096I

All other messages have been included because they were new for DL/I DOS/VS 1.9.

For messages *not* listed in the following, refer to the manual *DL/I Messages and Codes*.

DLZ1028   INVALID BUFFER SPECIFICATION IN HDBFR OPERAND, IT WAS SET TO *nn*

Explanation: The number of buffers specified in the HDBFR operand of the DLZACT TYPE=BUFFER macro instruction was not a numeric value from 2 to 255. It is set to the number indicated in the message text.

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.

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.

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.

DLZ1061   PSBLOC=<i>operand</i> **INVALID. PSBLOC=BELOW ASSUMED**

Explanation: The default value of PSBLOC=BELOW was assumed for the PSBLOC parameter in the DLZACT TYPE=CONFIG statement. PSBs will be loaded into storage below the 16MB line during CICS DL/I initialization.

DLZ1062   SVA=<i>operand</i> **INVALID. SVA=NO ASSUMED**

Explanation: The default value of SVA=NO was assumed for the SVA parameter in the DLZACT TYPE=CONFIG statement. DL/I phases will not be loaded into the SVA.

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*.

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.
8	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.

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".

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.

DLZ019I   BUFFER NUMBER IN {HDBFR|HSBFR} STATEMENT TOO {LOW|HIGH}, SET TO {MINIMUM|DEFAULT}

Explanation: One of the buffer number specifications did not fulfill the minimum requirements, or the specification was too high. DL/I resets the values to the minimum (if it was too low) or the default. See *DL/I DOS/VS Resource Definition and Utilities* for information on the requirements.

User response: None.

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.

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.

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.

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.

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.

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" on page 31.)

  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" on page 31.)

  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.

DLZ093I   **MPS STARTED WITH APPLID=**xxxxxxxx

Explanation: The Master Partition Controller (MPC) for DL/I Multiple Partition Support (MPS) started successfully. Batch jobs for MPS can now be started. xxxxxxxx denotes the CICS (generic) applid of the CICS/DLI partition.

User response: None required.

DLZ094I   **MPS WITH APPLID=**xxxxxxxx **STOPPED {ABNORMALLY | NORMALLY}**

Explanation: Multiple Partition support (MPS) ended abnormally or normally. If it ended abnormally the previous message gives the reason. No further batch jobs can be run in this CICS/MPS environment. xxxxxxxx denotes the CICS (generic) applid of the CICS/DLI partition.

User response: None required.

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).

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:

- 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.

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:

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

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.

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.

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.

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.

DLZ143I   MPS BATCH CONNECT REQUEST FOR {PARTID=*yy* | APPLID= *xxxxxxxx* |
PARTID=*yy*,APPLID=*xxxxxxxx*}

Explanation: An MPS batch job has been started. DL/I calls will be executed in the CICS/DLI MPS system as shown in the PARTID/APPLID parameter of the message text. *yy* denotes the VSE partition id, *xxxxxxxx* the CICS (generic) applid of the CICS/DLI partition.

User response: None required.

DLZ144I   CONTROL BLOCK DLZMPS OVERLAID

Explanation: The DL/I control block DLZMPS has been overwritten. This is an internal error. No more DL/I MPS batch jobs can be started. All active CICS/DLI MPS systems should be stopped (DL/I Stop transaction CSDD).

User response: Take a dump of the supervisor and contact your IBM Support Center.

DLZ145I   MORE THAN ONE MPS ACTIVE, SPECIFY 'PARTID=' AND/OR 'APPLID='

Explanation: An MPS batch job without a PARTID/APPLID parameter has been submitted, but more than one CICS/DLI MPS system is active. MPS cannot determine, to which of these systems it should connect.

User response: Specify an according PARTID and/or APPLID parameter in the DLI/DLR statement of the MPS batch job. Else ensure that only one CICS/DLI MPS system is active.

DLZ146I   RESTART PARTID/APPLID COULD NOT BE VERIFIED

Explanation: An MPS batch job has been restarted. The restarted MPS batch job expects the same CICS/DLI online environment as at the time of its abend. Therefore it will try to connect to the same partition and same applid. Either no CICS/DLI MPS system is active in this partition, or the CICS applid has changed. Message DLZ147A will follow.

User response: See message DLZ147A.

DLZ147A   ENTER 'CANCEL' TO END, OR NEW 'PARTID=' AND 'APPLID=' PARAMETER

Explanation: Message DLZ146I has been written before. The job can either be cancelled or should be executed in a different CICS/DLI MPS environment.

User response: Enter 'CANCEL', if you do not want to correct the error now. The job can then be restarted at another time. If execution should continue, a CICS/DLI MPS system must be active with the same DL/I MPS characteristics as at the time of the last execution. This includes the same Temporary Storage Queue (TSQ) and all required databases. When the correct CICS/DLI MPS system has been identified, enter:

```
PARTID=yy,APPLID=xxxxxxxx
```

yy denotes the VSE partition id, *xxxxxxxx* the CICS (generic) applid of the CICS/DLI MPS system, where the job should be restarted.

DLZ250I   CICS REGISTRATION CALL FAILED, DL/I TASK TERMINATED

Explanation: A CICS transaction tried to register as a DL/I task, but the registration failed.

User response: This is an internal error, which inhibits CICS/DLI online operation. Report the problem to your IBM support center.

DLZ251I   DL/I NOT ACTIVE, TASK TERMINATED

Explanation: A CICS transaction issued a DL/I call, but the DL/I online system is not active. Errors have been encountered during CICS/DLI online initialization.

User response: Correct the errors. Ensure that message DLZ069I is shown during CICS/DLI online initialization.

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.

DLZ431I   **WARNING - RECORDS ON THE LOGFILE SEEM TO BE INCOMPLETE**

Explanation: An open record on the input log file is missing. The log file or log file sequence may be incomplete.

User response: Ensure that the set of input logfile(s) is complete. Resubmit the job, if necessary.

New DL/I CICS Return Codes

Type of Call	UIBFCTR UIBDLTR	Explanation
Scheduling	0C 09	Internal error at scheduling conflict; contact IBM support

Appendix A. Sample Job Streams for DL/I 31-Bit Applications

This chapter contains sample job streams showing, how to create DL/I 31-bit applications for the different environments, using different programming interfaces and languages.

DL/I - Online Applications

For DL/I online applications written in COBOL for VSE/ESA the job control requirements are the same as for DL/I online applications written in COBOL II.

For DL/I online applications written in PL/I for VSE/ESA the formerly required inclusion of object module DFHPL1I must be replaced through the inclusion of DFHELII. The inclusion of object module PLISHRE must be dropped. All programs should be compiled with option SYSTEM(CICS).

DL/I online applications written in High Level Assembler for VSE can be assembled and link-edited as in the previous way.

COBOL Online Example:

The following example illustrates the job control statements needed to compile and link a DL/I online application written in COBOL for VSE/ESA or COBOL II. IGYCRCTL is the name for both compilers. If both compilers are installed the LIBDEF SEARCH chain determines, which compiler will actually be used.

```
// JOB CBLSAMPL
// DLBL IJSYSPH,'COBOL TRANSLATION',yyyy/ddd
// EXTENT SYSPCH,extent information
ASSGN SYSPCH,DISK,VOL=volid,SHR
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// EXEC DFHECP1$,SIZE=...
 CBL XOPTS(CICS,DLI,COBOL2),LIB,RES,RENT,DATA(31)
        .
        .
    SOURCE DECK
        .
        .
```

Figure 15. COBOL Online with HLPI or CALL Interface (Part 1 of 2).

```
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'COBOL TRANSLATION',yyyy/ddd
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// OPTION NODECK,CATAL
    PHASE CBLSAMPL,*
    INCLUDE DFHELII
// EXEC IGYCRCTL,SIZE=...
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&
```

Figure 15. COBOL Online with HLPI or CALL Interface (Part 2 of 2).

Note: When the DL/I CALL interface is used the 'DLI' parameter in the CBL
XOPTS statement should be omitted.

As an alternative to DFHELII, the interface DFHECI can be selected.

PL/I for VSE/ESA Online Example:

The following example illustrates the job control statements needed to compile and link a DL/I online application written in PL/I for VSE/ESA.

```
// JOB PLISAMPL
// DLBL IJSYSPH,'PL/I TRANSLATION',yyyy/ddd
// EXTENT SYSPCH,extent information
ASSGN SYSPCH,DISK,VOL=volid,SHR
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// EXEC DFHEPP1$,SIZE=...
*PROCESS SYSTEM(CICS),XOPTS(CICS,DLI)
      .
      .
   SOURCE DECK
      .
      .
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'PL/I TRANSLATION',yyyy/ddd
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// OPTION NODECK,CATAL
   PHASE PLISAMPL,*
   INCLUDE DFHELII
// EXEC IEL1AA,SIZE=...
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&
```

Figure 16. PL/I Online with HLPI or CALL Interface

Note: When the DL/I CALL interface is used the ′DLI′ parameter in the *PROCESS XOPTS statement should be omitted.

High Level Assembler Online Example:

The following example shows the job control statements needed to assemble and link a DL/I online application written in High Level Assembler for VSE.

```
// JOB HLASAMPL
// DLBL    IJSYSPH,'ASSEMBLER TRANSLATION',yyyy/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=...
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&
```

Figure 17. Example: High Level Assembler - Online

Note: DL/I online applications written in High Level Assembler for VSE must be implemented using the DL/I CALL interface.

DL/I - Batch Applications

For DL/I batch and batch/MPS applications written either in COBOL for VSE/ESA or COBOL II the inclusion of object module DLZBPJRA must be dropped. Therefore the JCL definition for the former entry point CBLCALLA must be replaced by the new entry point DLITCBL.

For DL/I batch and batch/MPS applications written in PL/I for VSE/ESA, the inclusion of IBMBPJRA must be replaced through the inclusion of IBMRPJRA. Also all programs should be compiled with option SYSTEM(DLI).

DL/I batch applications written in High Level Assembler for VSE can be assembled and link-edited in the same way as before.

COBOL Batch Examples:

The following examples illustrate the job control statements needed to compile and link a batch or batch/MPS application written either in COBOL for VSE/ESA or COBOL II language. IGYCRCTL is the name for both compilers. If both compilers are installed the LIBDEF SEARCH chain determines, which compiler will actually be used.

```
// JOB COBSAMPL
// DLBL IJSYSPH,'COBOL TRANSLATION',yyyy/ddd
// EXTENT SYSPCH,extent information
ASSGN SYSPCH,DISK,VOL=volid,SHR
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// EXEC DFHECP1$,SIZE=...
 CBL XOPTS(DLI,COBOL2),RES,RENT,DATA(31)
        .
        .
   SOURCE DECK
        .
        .
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'COBOL TRANSLATION',yyyy/ddd
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// OPTION NODECK,CATAL
   PHASE COBSAMPL,*
   INCLUDE DLZLICBL
// EXEC IGYCRCTL,SIZE=...
   ENTRY DLITCBL
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&
```

Figure 18. COBOL Batch and Batch/MPS with HLPI Interface

31-Bit Applications

```
// JOB COBSAMPL
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// OPTION NODECK,CATAL
   PHASE COBSAMPL,*
// EXEC IGYCRCTL,SIZE=...
 CBL LIB,RES,RENT,DATA(31)
        .
        .
   SOURCE DECK
        .
        .
/*
   ENTRY DLITCBL
// EXEC LNKEDT
/&
```

Figure 19. COBOL Batch and Batch/MPS with CALL Interface

PL/I for VSE/ESA Batch Examples:

The following examples illustrate the job control statements needed to compile and link a batch or batch/MPS application written in PL/I for VSE/ESA.

```
// JOB PLISAMPL
// DLBL IJSYSPH,'PL/I TRANSLATION',yyyy/ddd
// EXTENT SYSPCH,extent information
ASSGN SYSPCH,DISK,VOL=volid,SHR
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// EXEC DFHEPP1$,SIZE=...
*PROCESS SYSTEM(DLI),XOPTS(DLI)
        .
        .
   SOURCE DECK
        .
        .
/*
CLOSE SYSPCH,punch
// DLBL IJSYSIN,'PL/I TRANSLATION',yyyy/ddd
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volid,SHR
// OPTION NODECK,CATAL
   PHASE PLISAMPL,*
// EXEC IEL1AA,SIZE=...
   INCLUDE DLZLIPLI
   INCLUDE IBMRPJRA
   ENTRY DLZLIPLI
// EXEC LNKEDT
/&
// JOB RESET
CLOSE SYSIPT,reader
/&
```

Figure 20. PL/I for VSE/ESA Batch and Batch/MPS with HLPI Interface

```
// JOB PLISAMPL
// LIBDEF *,SEARCH=search-libraries
// LIBDEF PHASE,CATALOG=catalog-library
// OPTION NODECK,CATAL
   PHASE PLISAMPL,*
// EXEC IEL1AA,SIZE=...
*PROCESS SYSTEM(DLI)
        .
        .
   SOURCE DECK
        .
        .
/*
   INCLUDE IBMRPJRA
   ENTRY PLICALLB
// EXEC LNKEDT
/&
```

Figure 21. PL/I for VSE/ESA Batch and Batch/MPS with CALL Interface

Note: The former PL/I ISA storage is now controlled by LE/VSE, which uses GETVIS for storage allocation. This can lead to an increased demand for GETVIS storage of your applications.

High Level Assembler for VSE Batch Example:

The following example shows the job control statements needed to assemble and link-edit a DL/I batch application written in High Level Assembler for VSE.

31-Bit Applications

```
// JOB HLASAMPL
// LIBDEF *,SEARCH=search-library
// LIBDEF PHASE,CATALOG=catalog-library
// OPTION CATAL,NODECK
   PHASE HLASAMPL,*
// EXEC ASMA90,SIZE=...
HLASAMPL CSECT
HLASAMPL AMODE 31
HLASAMPL RMODE ANY
        .
        .
   SOURCE DECK
        .
        .
/*
// EXEC LNKEDT
/&
```

Figure 22. Example: High Level Assembler - Batch and MPS Batch

Appendix B. Summary of Customer Interfaces

This appendix contains General-use Programming Interface and Associated Guidance Information, and Product-sensitive Programming Interface and Associated Guidance Information.

DL/I Macros Intended for Customer Use

Table 3 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.

Table 3. Summary of Customer Interfaces

Macro Name	General Use	Product Sensitive	Described in (see Legend below)
ACCESS	x		RDU, IRDU, GNU
ALLDLI	x		DIAG, CALL
ATASET	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

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*

Customer Interfaces

Related IBM Manuals

DL/I VSE 1.11

For the latest information on DL/I, the DL/I VSE 1.11 Release Guide is the first manual you should consult.

The following lists all the manuals of the DL/I library. In the list below, "(1)" means that the manual is referenced in this manual.

DL/I General Information, GH20-1246-9

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

DL/I Licensed Program Specifications, GH24-5031-4

DL/I Release Guide, SC33-6211-5 (1)

DL/I Guide for New Users, SH24-5001-4 (1)

DL/I Application and Data Base Design, SH24-5022-1

DL/I Data Base Administration, SH24-5011-1 (1)

DL/I Resource Definition and Utilities, SH24-5021-2 (1)

DL/I Interactive Resource Definition and Utilities, SH24-5029-0 (1)

DL/I Application Programming: High Level Programming Interface, SH24-5009-2 (1)

DL/I Application Programming: CALL and RQDLI Interfaces, SH12-5411-6 (1)

DL/I Messages and Codes, SH12-5414-10 (1)

DL/I Diagnostic Guide, SH24-5002-4 (1)

DL/I Recover and Restart Guide, SH24-5030-0

IBM System/370 LLC/CC in DL/I DOS/VS Program Reference and Operations Manual, SH20-9046-3

Summaries:

DL/I Reference Summary: Application Programming, SX24-5103-4

DL/I Reference Summary: System Programming, SX24-5104-4

DL/I Reference Summary: High Level Programming Interface, SX24-5120-2

Logic Manuals:

DL/I Logic, Volume 1, LY12-5016-7

DL/I Logic, Volume 2, LY12-5015-1

DL/I Logic Extensions, LY33-9123-1 (1)

Selected VSE Manuals

z/VSE Planning, SC33-8221

z/VSE Installation, SC33-8222

z/VSE Operation, SC33-8239

z/VSE System Control Statements, SC33-8225

VSE/ESA Extended Addressability, SC33-6621

Related Manuals

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.

CICS TS (Customer Information Control System Transaction Server). An IBM program that controls online communication between terminal users and a data base. Transactions entered at remote terminals are processed concurrently by user-written application programs. The program includes facilities for building, using, and servicing data bases.

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.

DATABASE 2. An IBM relational database management system.

DB2. See DATABASE 2.

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.

CSD (CICS system definition) file. A component of CICS resource definition online (RDO). It keeps a permanent record of resource information, independently of the active CICS system. The information held in the CSD is used for installing new resources and at a CICS restart.

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 z/VSE.

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.

Z

z/VSE. The most advanced VSE system currently available.

Index

Numerics

31-bit applications   12

A

APAR enhancements   8
applications, 31-bit   12

B

batch applications   81

C

CICS Transaction Server 1.1, new with
  VSE/ESA 2.4   1
CICS/VSE 2.3   1
CICS/XRF environment   59
COBOL for VSE/ESA   12
COBOL II   12
Coexistence Environment   18
customizing DL/I   31

D

DBD definition, example   4
DFHECI   78
DFHELII   78
DL/I DOS/VS 1.10 enhancements   11
DL/I enhancements, latest (VSE/ESA
  2.4)   1
DL/I in a VM environment   20
DL/I VSE 1.11, new with VSE/ESA
  2.4   1
DLZMPI00   2
DLZRRC00   2
DLZUACB0 block builder utility   3
DLZURGL0 reload utility   4
DLZURGU0 unload utility   4
dynamic partition support   57

E

enhancements introduced via APARs   8
enhancements, DL/I DOS/VS 1.10   11
enhancements, DL/I VSE 1.11   1

G

GETVIS, for program isolation
  storage   14

H

HD buffers, more than 32   15
HD databases, greater 4GB   3
High Level Assembler   12

HS buffers, VSE/VSAM   62

I

index database, increased VSAM
  blocksize   13
installation dialogs   23
installation tapes   23
installing DL/I   23

M

macros for customer use   85
messages   67
messages, new and changed   18
migration items, CICS Transaction
  Server   17
migration items, DL/I   16
MPS batch job   1
Multiple MPS Systems   1

N

new messages   67

O

online applications   77

P

PL/I for VSE/ESA   12
post installation tasks   41
PSB, above 16 MB   15

S

shipment of DL/I   23
shipment tapes   23
storage protection for CICS Transaction
  Server   7
SVA loading, conditional   7
SVA, loading phases into   31

T

TRAP runtime option   12

V

virtual disk   63
VSE/ESA enhancements   1

Y

year 2000 ready   viii
year 2000 support   14

Z

z/VSE home page   ix

Readers' Comments — We'd Like to Hear from You

IBM Data Language/I
Virtual Storage Extended
Release Guide
Version 1 Release 11

Publication No. SC33-6211-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?   ☐ Yes   ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

IBM ®

Fold and Tape   **Please do not staple**   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Attn: Dept EHJ - BP/003D
6300 Diagonal Highway
Boulder, CO
 80301-9151

Fold and Tape   **Please do not staple**   Fold and Tape

IBM.

Spine information:

IBM

IBM Data Language/I
Virtual Storage Extended

IBM DL/I Release Guide

Version 1 Release 11