IMS Fundamentals

(Course Code CM01)

Student Notebook

ERC 1.

IBM Education and Training

Publishing Information

This publication has been produced using BookMaster (Program Number 5668-015), the Document Composition Facility (Program Number 5748-XX9), and Freelance 2.1 for Windows. It was printed on the IBM 3820 Page Printer.

First Edition (December, 1996)

Contents

Course Description

IMS Fundamentals

Duration : 3.0 days

Purpose

This course is designed to present an introduction to the basic facilities of IMS, how these facilities work together, and how application programs interact with them. The course presents the IMS Database System, the IMS Transaction Manager, and their use in today's modern Enterprise Systems.

Audience

Users who want a basic understanding of the IMS Database and IMS Transaction Manager products and data processing personnel who will be working with an IMS System (DB, DB/DC, DBCTL.).

Prerequisites

Before taking this course, you should be able to describe the fundamentals of data processing, including the function of operating systems, access methods and Job Control Language (JCL). These skills can be obtained from on-the-job experience.

Objectives

- List the elements of the IMS Database system
- Identify the IMS database organizations.
- Describe the processing of a database record in hierarchic sequence.
- Match the DBD and PSB control blocks used in database processing with their function.
- Contrast the requirements and effects of sequential versus direct access of databases.
- Select the reasons for use of secondary indexes and logical relationships
- Compare full-function and Fast Path database organization

- List the basic IMS Data Communictions functions of the IMS Transaction Manager (TM)
- Identify the roles of messages, queues, and logical terminal names
- Distinguish among the scheduling characteristics of the IMS/TM region types: Message Processing programs, Batch Message Processing programs, and Interactive Fast Path
- State the role of commit points in recovery and restart
- Identify the functions of dynamic backout, batch backout, the system log, and the benefits of periodically backing up a database.
- Cite the capabilities of a conversational program and its implications on processing and performance.
- Describe the functions of additional IMS product features including DBRC, IRLM, Data Sharing, Distributed Processing, and DBCTL for CICS

Contents

- A mature product world-wide:
  - Over 25 years on the market, over 7000 installations
  →Widely-used and accepted by large corporations
  - Most critical corporate data under IMS
- Two separate strategic products:
  1. IMS/ESA™ Database Manager:
     - Full-Function databases
     - Fast Path databases
     - Logging and Recovery Functions
  2. IMS/ESA™ Transaction Manager:
     - Data-communications "interactive" online functions
     →Makes data available to the end-user...
     - Transaction-driven
     - Logging and Integrity

Agenda

Day 1

Welcome
Administration
IMS Overview
IMS Database Manager (DBM)
Lunch
IMS Database Handling (DBH)
Hierarchic Access Methods (HAM)
IMS Other DL/I Functions (DBO)

Day 2

LAB 1 and Review
IMS Transaction Manager System (TMS)
LUNCH
IMS Transaction Message Processing (TMP)
IMS Fast Path (IFP)

Day 3

Lab 2 and Review
IMS Large-System Environment (LSE)
-CICS with DBCTL -Data Sharing
IMS Distributed Processing (DPR)
Lunch
IMS-DB2 Attachment (DB2)

Unit 1. IMS Database Manager

What This Unit Is About

How the IMS/DB software is used to manage an organization's information

What You Should Be Able to Do

After completing this unit, you should be able to:

- List the elements of an IMS Database System
- Describe the processing of a database in hierarchic sequence
- Identify the principal control blocks used by IMS/DB

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 1 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8012-00 IMS/ESA V5 Administration Guide: DB

SC26-8034-00 IMS/ESA V5 Utilities Reference: DB

1.1 IMS Database Manager

What Is A Database System?

"A SYSTEM THAT ALLOWS
MULTIPLE INDEPENDENT USERS
TO HAVE CONCURRENT ACCESS
TO A CENTRAL REPOSITORY
OF INFORMATION..."

- ADVANTAGES

1. Centralized files for all applications
2. Elimination of duplicate space and effort
3. Single information source provides complete, accurate information processing

Figure 1-1. What is a Database System (CM011103)

Notes:

A Central Repository

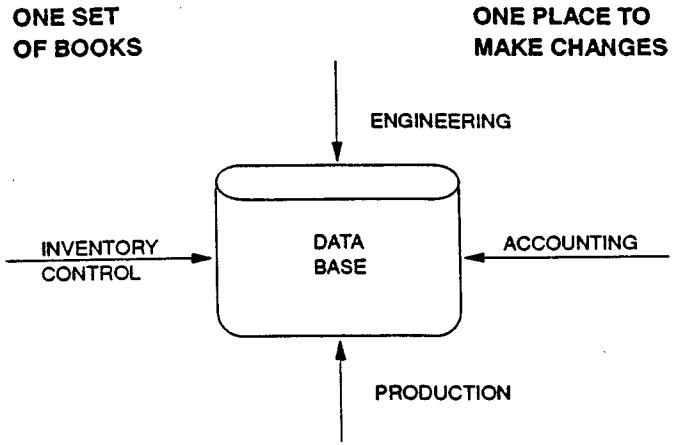


Figure 1-2. A Central Repository (CM011105)

Notes:

What Is DL/I?

**"DATA LANGUAGE/ONE IS A
DATA MANAGEMENT FACILITY
THAT SERVES AS AN INTERFACE
BETWEEN AN APPLICATION PROGRAM
AND A DATABASE..."**

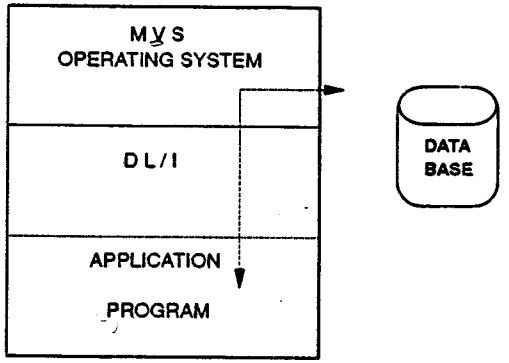
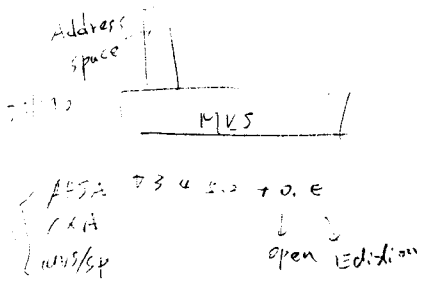


Figure 1-3. What is DL/I ? (CM011107)

Notes:



Limits of Traditional Systems

- Fixed file contents limits the use of the data, and impacts application expansion
- Additional files create data redundancy, and additional data maintenance
- Interrelating data from multiple files complicates design and programming
- Recovery procedures are non-standard, or ignored until a disaster occurs!

Figure 1-4. Limits of Traditional Systems (CM011109)

Notes:

Traditional Approach

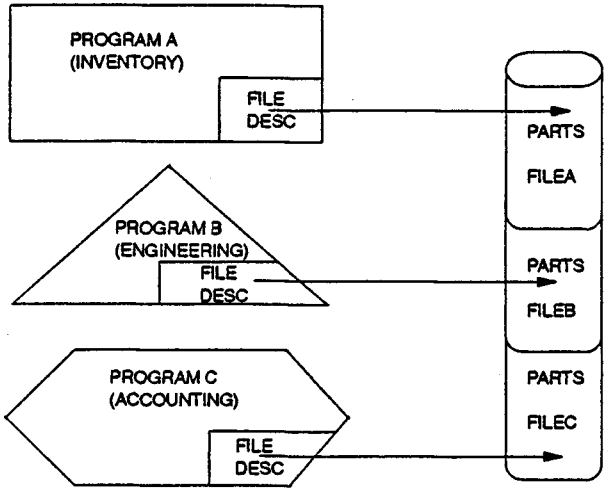


Figure 1-5. Traditional Approach (CM011111)

Notes:

Database Approach

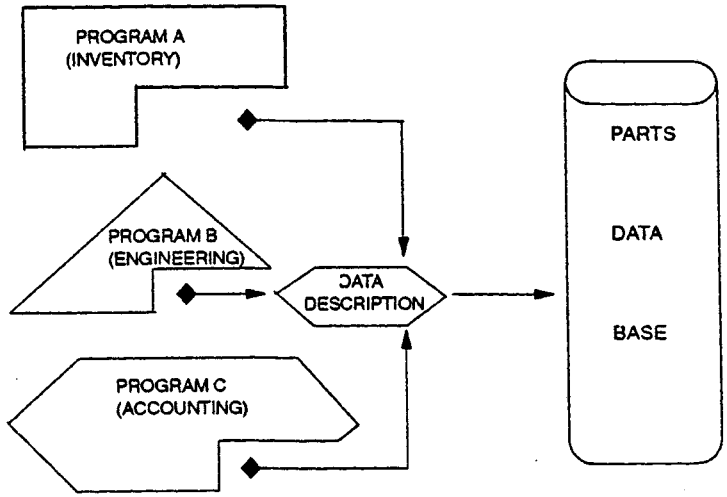


Figure 1-6. Database Approach (CM011113)

Notes:

Fixed File Contents

EMPLOYEE MASTER Record:

Field	Length	Field	Length
Employee No.	6	Taxes	
Employee Name	30	Fit	6
Department	2	FICA	5
Date Hired	6	State	5
Address Line 1	25	Local	5
Address Line 2	25	Deductions (Up to 5)	
Address Line 3	25	Type 2	
Address Line 4	25	Amount 5	
Gross Salary	6	Balance 5	
Salary Class	3	(12 x 5)	60
Auth Date	6		

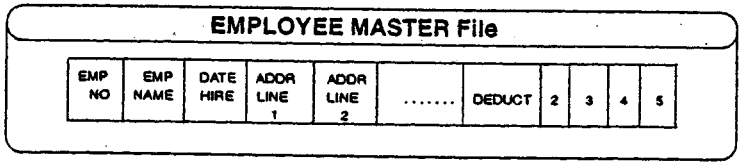


Figure 1-7. Fixed file Contents (CM011115)

Notes:

DL/I Hierarchy

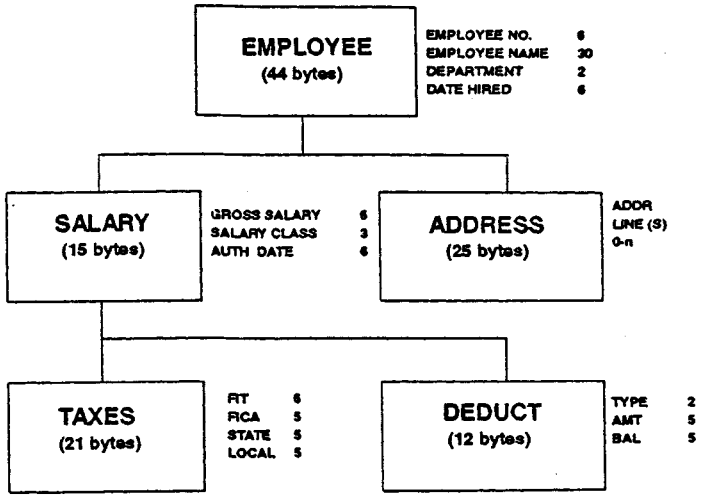


Figure 1-8. DL/I Hierarchy (CM011117)

Notes:

ROOT: The **highest-level** (no parent) segment in a hierarchy

PARENT: A Segment **above** a dependent segment

CHILD: A Segment **below** and dependent on a higher-level segment

There is a maximum of 255 segment **types**

DL/I Hierarchy...Cont.

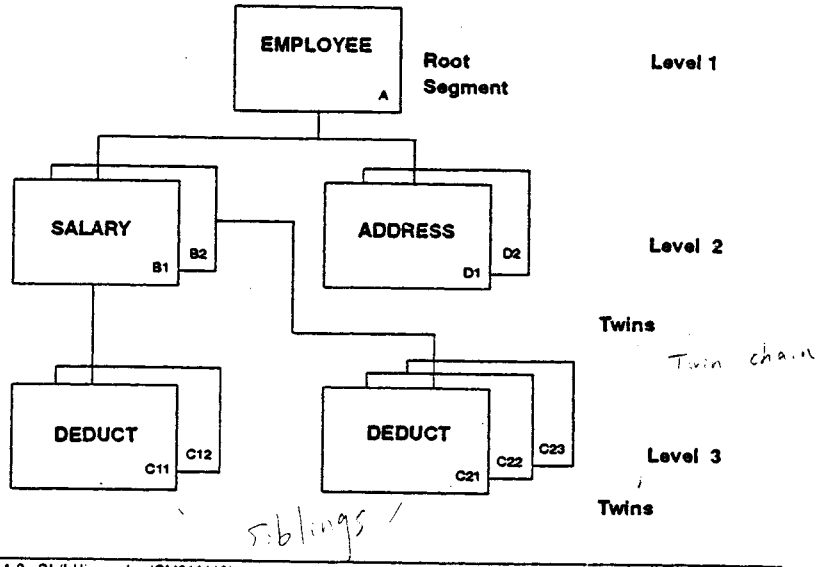


Figure 1-9. DL/I Hierarchy (CM011119)

Notes:

Twins: Multiple occurrences of the same segment type under the same parent occurrence

Levels: The position of segments in relation to the root and other segments. There is a maximum of 15 segment hierarchic levels.

Database Record: A Root Segment and all of its dependents. A database consists of 1-n database records.

Hierarchic Sequence

EXERCISE: Place the number or letter in the boxes in the sequence that these segments will be accessed by a sequential "get-next" function.

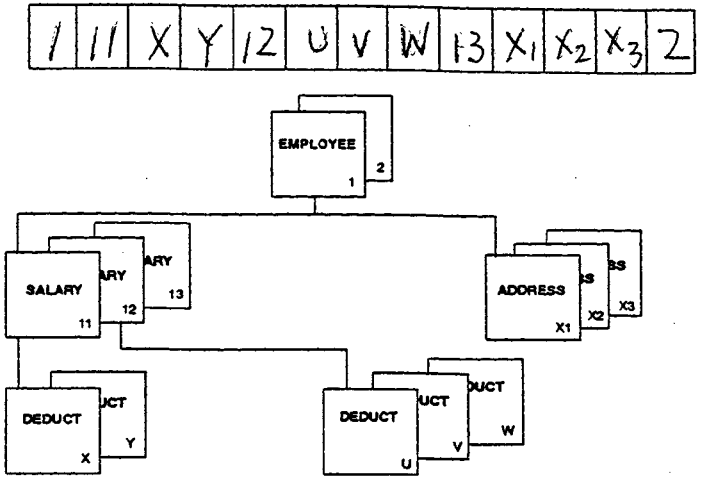


Figure 1-10. Hierarchic Sequence (CM011121)

Notes:

Sequential Processing Order 1)Top-to-bottom 2)Front-to-back 3)Left-to-right

Add New Application

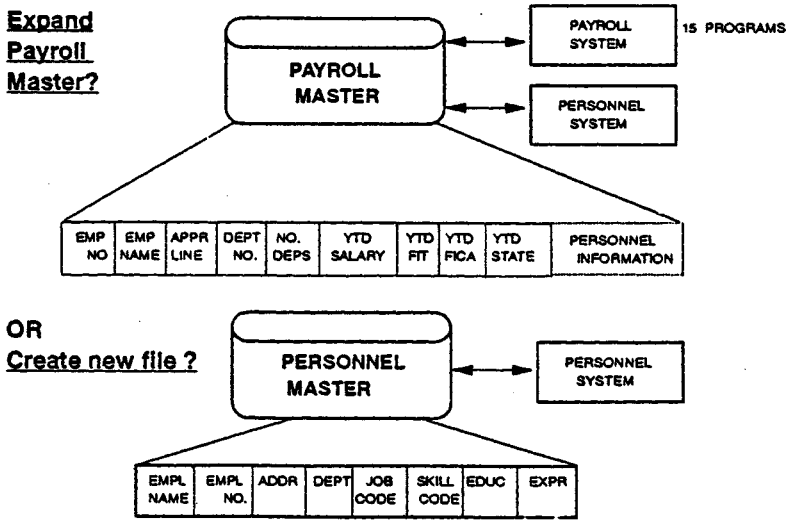


Figure 1-11. Add New Application (CM011123)

Notes:

DL/I Easy To Expand

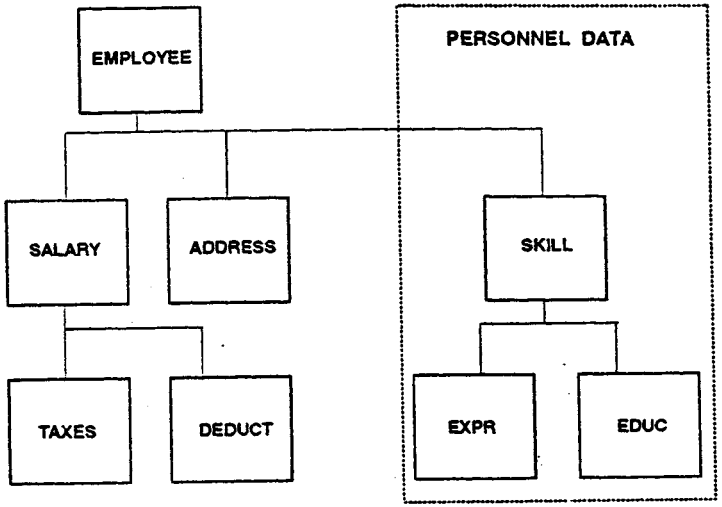
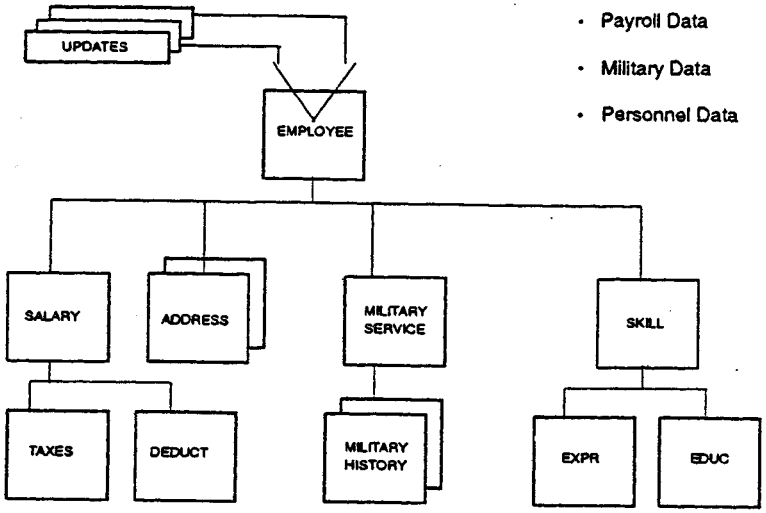


Figure 1-12. DL/I Easy To Expand (CM011125)

Notes:

Multi-Application Updates



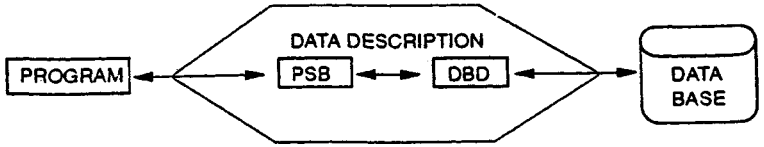
- Payroll Data
- Military Data
- Personnel Data

Figure 1-13. Multi-Application Updates (CM011127)

Notes:

Data Independence

**SEPARATE THE PROGRAM
FROM THE PHYSICAL CHARACTERISTICS
OF THE DATABASES:**



- ADVANTAGES:

1. Simplifies application program development
2. Provides security, integrity, and consistency of a database
3. Facilitates changes to database

Figure 1-14. Data Independence (CM011129)

Notes:

PSB = Program Specification Block: Program view of the databases

DBD = Database Description: Physical characteristics of the databases

Logical View/Physical View

Logical View/Physical View

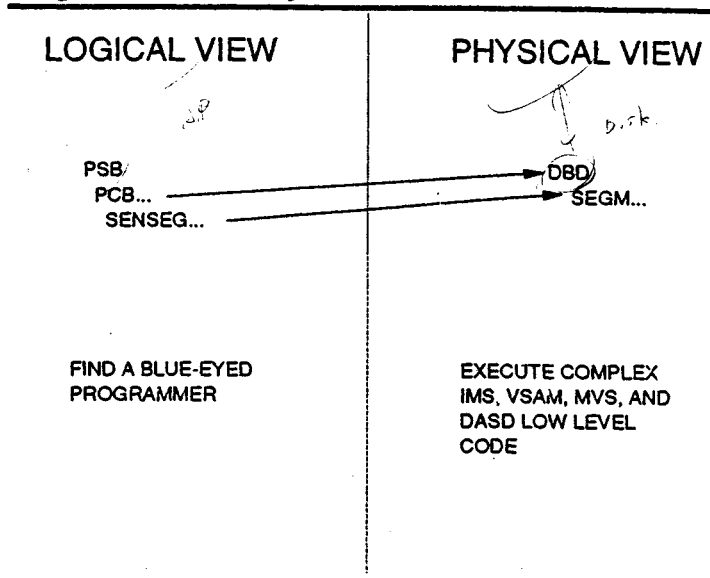


Figure 1-15. Logical View/Physical View (CM011131)

Notes:

Database Description (DBD)

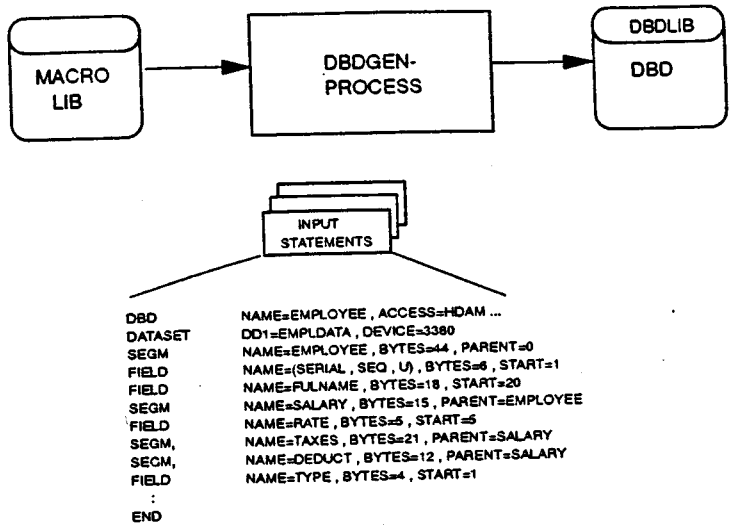


Figure 1-16. Database Description (DBD) (CM011133)

Notes:

Program Specification Block (PSB)

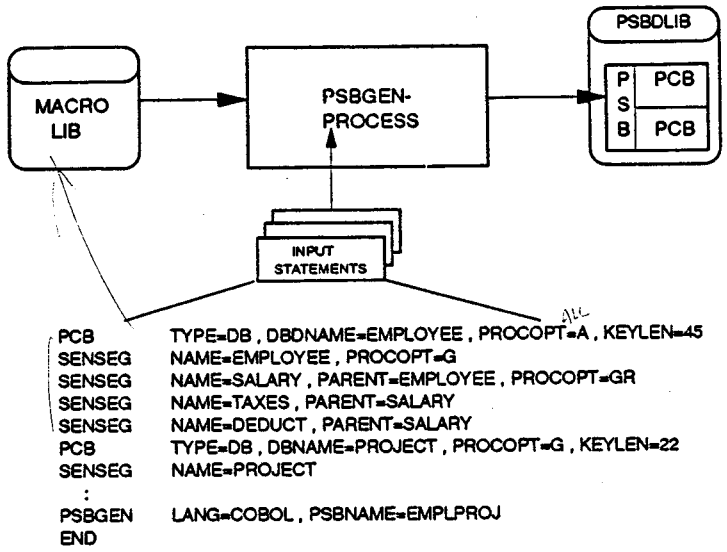
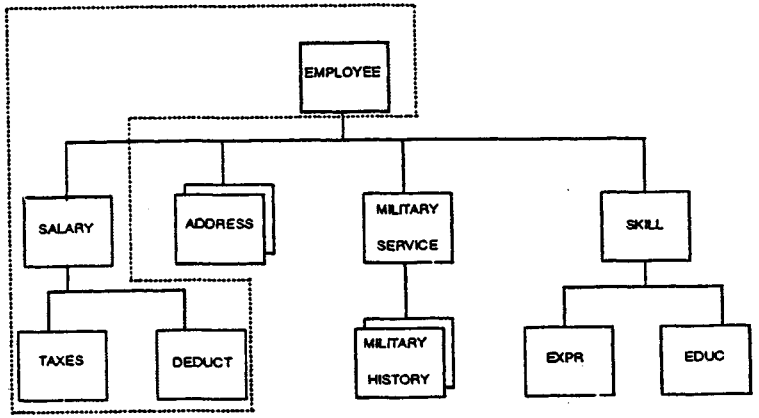


Figure 1-17. Program Specification Block (PSB) (CM011135)

Notes:

PCB = Program Communication Back

Segment Sensitivity



Logical Data Structure for Payroll processing

What is Logical Data Structure for an Employee Mailing List?

Figure 1-18. Segment Sensitivity (CM011137)

Notes:

Batch IMS Execution

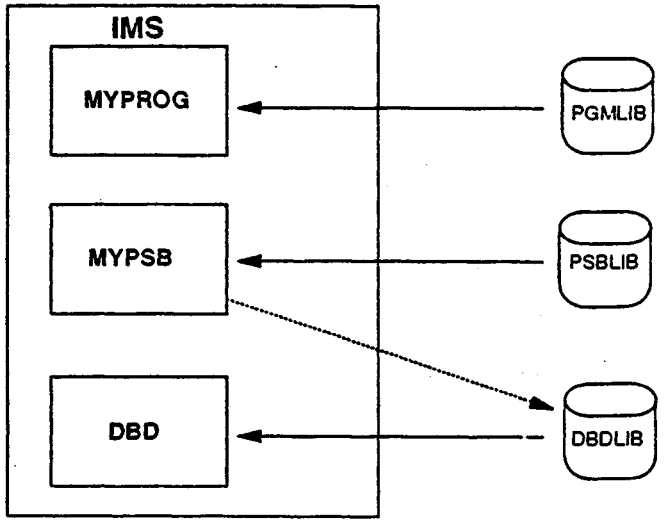
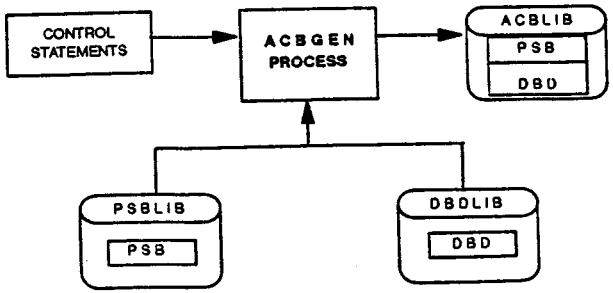


Figure 1-19. Batch IMS Execution (CM011139)

Notes:

//EXEC PGM=DFSRRC00, PARM='DLI, MYPROG, MYPSB'

Application Control Blocks



- Application Control Blocks (ACB) Utility:
  - Defines Application Control Block library (ACBLIB)
  - Uses database descriptors (DBDs)
  - Uses program descriptions (PSBs)
- ACBGEN:
  - Verifies PSB-PCB-DBD existence and compatibility
  - Verifies KEYLEN parameter

Figure 1-20. Application Control Blocks (CM011141)

Notes:

Application Program View

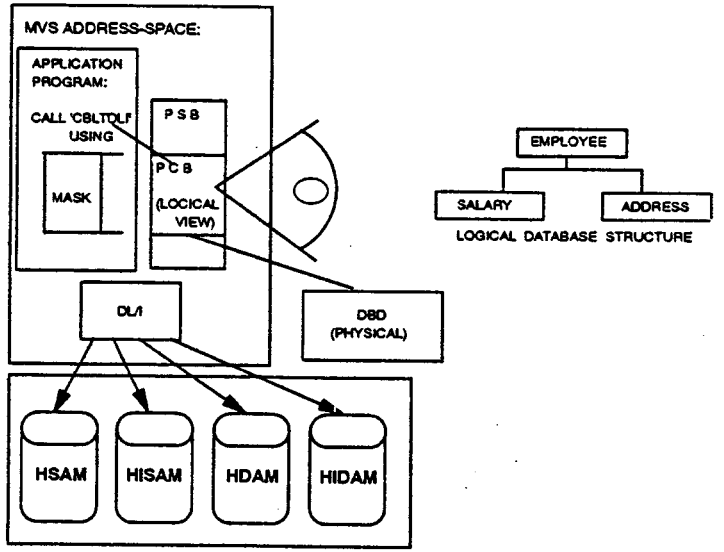
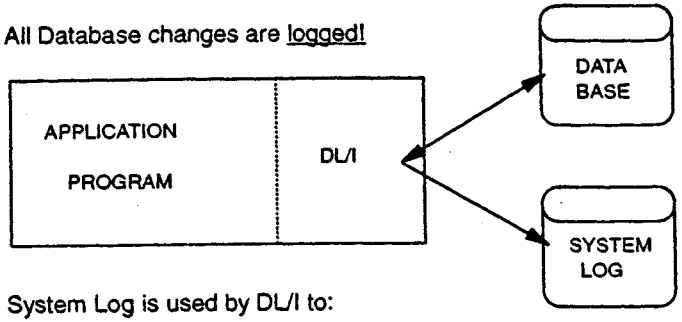


Figure 1-21. Application Program View (CM011143)

Notes:

Database Recovery

- All Database changes are logged!

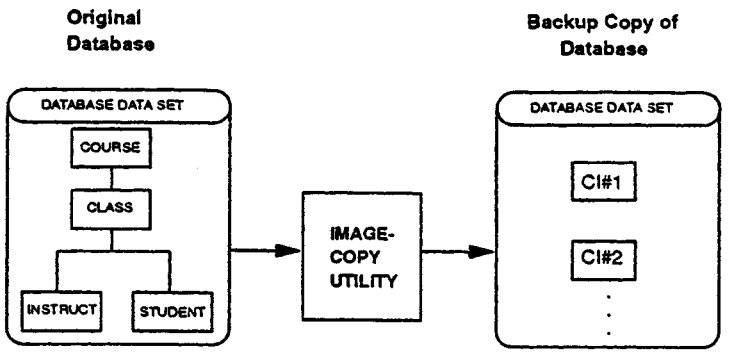


- System Log is used by DL/I to:
  1. Recover databases from errors
  2. Back out updates when program abnormally terminates
- "Before" and "After" images of DL/I segments

Figure 1-22. Database Recovery (CM011145)

Notes:

Backing-Up A Database



CIC - CONCURRENT IMAGE COPY ALLOWS UPDATE OF DATABASES WHILE IMAGE COPY RUNS. PRODUCES "FUZZY" IMAGE COPY.

Figure 1-23. Backing-Up A Database (CM011149)

Notes:

The Image-Copy header record contains a timestamp of when it is created.

Logging Database Changes

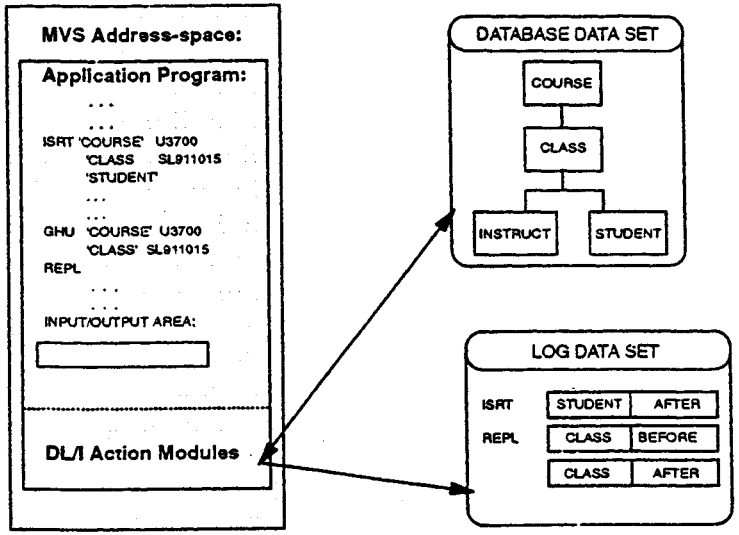


Figure 1-24. Logging Database Changes (CM011147)

Notes:

Each log record has a timestamp to identify it.

Database Recovery

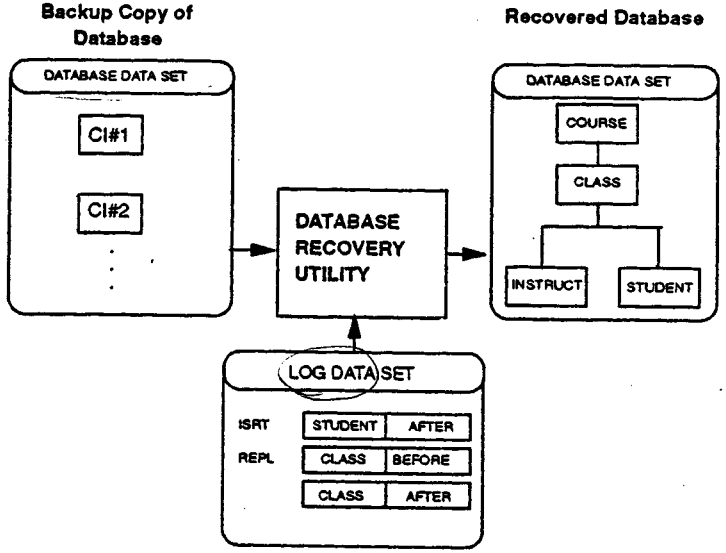
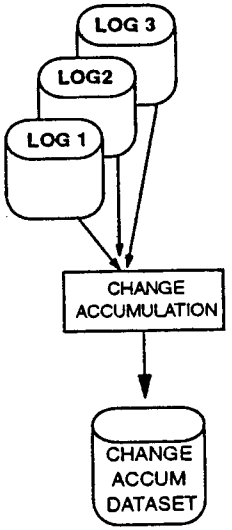


Figure 1-25. Database Recovery (CM011151)

Notes:

1. Original databases was backed-up, but STUDENT and CLASS have been subsequently updated.
2. Hardware error causes the database to become unusable !
3. The backup copy is out-date: it does not reflect changes to STUDENT and CLASS
4. Databases updates on log data set after time of backup are re-applied; each log record has a timestamp to identify it.
5. The "recovered" database is now in agreement with the original.

Database Log Change Accumulation



MUST BE IN CREATION SEQUENCE

OPTIONAL - INCLUDE PREVIOUS CHANGE ACCUM DATASET

SORT TO CUT RECOVERY TIME. CONSOLIDATE CHANGES

Figure 1-26. Database Log Change Accumulation (CM011153)

Notes:

DL/I Summary

DATA INDEPENDENCE:
- No file description coding within program
- No re-compilations due to changes in the physical storage or access method
- Same type of DL/I coding regardless of language, or physical access method

DATA INTEGRATION:
- Less file-related logic to code
- Only one field in one database to maintain

DATA SENSITIVITY:
- Less storage to reserve
- Fewer fields to define
- No re-compilations due to addition/deletion of non-sensitive segments
- Processing Options (PROCOPTs)

DATA BASE INTEGRITY:
- No logic or coding for recovery in case of failure !

Figure 1-27. Summary (CM011155)

Notes:

Unit 2. IMS Database Handling

What This Unit Is About

Application program structure to access IMS databases

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss the elements of a call to retrieve or update information in an IMS database.

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 1 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8015-01 IMS/ESA V5 Application Programming: DB

Objective

- Discuss the elements of a call to retrieve or update information in an IMS database.

Figure 2-1. Objective (CM011201)

Notes:

DL/I Processing

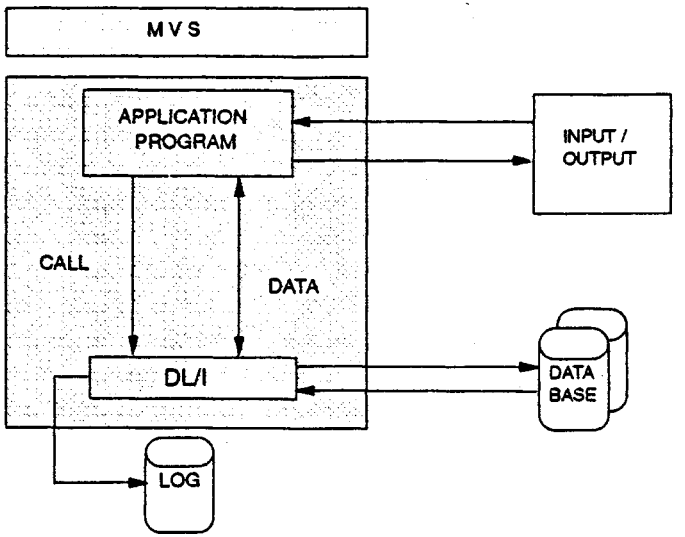


Figure 2-2. DL/I Processing (CM011200)

Notes:

Application Program Structure

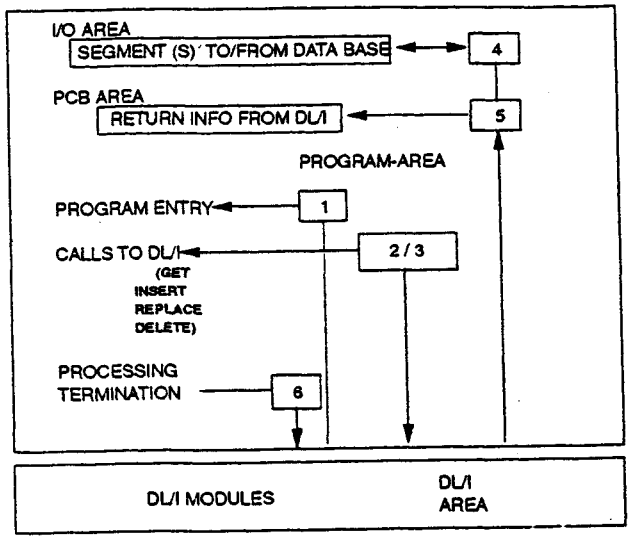


Figure 2-3. Application Program Structure (CM011203)

Notes:

Application Program Structure...

1. Application Program Entry
2. Function (Get, Insert, Replace, Delete)
3. Segment Search Arguments (SSAs):
   - Segment Name
   - Key for Search
   - Boolean Operators and/or Logic
   - Command Codes
     - → e.g., "Path" Calls
4. Input/Output Area in Application Program
5. Program Communication Block (PCB) contains:
   - → Database Name
   - → Area for Return Codes to Application Program
6. Termination

Figure 2-4. Application Program Structure ... (CM011205)

Notes:

Application Program Entry Point

1. COBOL:

```
PROCEDURE DIVISION USING DB-PCB1, DB-PCB2.
```

OR:

```
ENTRY 'DLITCBL 'USING DB-PCB1, DB-PCB2,
```

2. PL/I:

```
DLITPLI:  PROC (DB_PCB1, DB_PCB2) OPTIONS (MAIN);
```

OR:

```
DCL DLITPLI ENTRY,
    DB_PCB1 POINTER, DB_PCB2 POINTER;
```

3. ASSEMBLER:

Standard convention: Upon entry, register 1 points to a parameter list of PCB pointers

Example:

```
CSECT
L       9,0 (1)     LOAD PCB1 ADDRESS
USING   PCB1, 9
L       7, 4 (1)    LOAD PCB2 ADDRESS
USING   PCB2, 7
```

Figure 2-5. Application Program Entry Point (CM011207)

Notes:

Program Communication Block (PCB)

Generated by PSB Utility **Mask Written inCOBOL**

SIZE	FUNCTION	
		01 DB-PCB1.
8	NAME OF DATABASE	02 DBD-NAME PIC X(8).
2	SEGMENT HIERARCHY LEVEL	02 SEG-LEVEL PIC XX.
2	DL/I STATUS CODE	02 STATUS-CODE PIC XX.
4	DL/I PROCESSING OPTIONS	02 PROC-OPTIONS PIC XXXX.
4	RESERVED FOR DL/I	02 RESERVE-DLI PIC S9(5) COMP.
8	SEGMENT NAME FEEDBACK	02 SEG-NAME-FB PIC X(8).
4	LENGTH OF FEEDBACK KEY	02 LENGTH-FB-KEY PIC S9(5) COMP.
4	NUMBER OF SENSEGS	02 NUM-SENS-SEGS PIC S9(5) COMP.
		02 KEY-FB-AREA.
		03 SKILL-KEY PIC X(4).
n	KEY FEEDBACK AREA	03 NAME-KEY PIC X(5).
		03 EXPR-KEY PIC X(3).

Figure 2-6. Program Communication Block (PCB) (CM011209)

Notes:

ASM Example:

```
DBPCB     DSECT
DBPCBDB   DS      CL8
DBPCBLEV  DS      CL2
DBPCBSC   DS      CL2
DBOCBPRO  DS      CL4
DBPCBRSV  DS      F
DBPCBSEG  DS      CL8
DBPCBLKY  DS      F
DBPCBNSS  DS      F
DBPCBKFD  DS      Cln
```

PL/I Example:

```
DCL         PCB_POINTER          POINTER
DCL1        DB_PCB1              BASED (PCB_POINTER),
            2 BDB_NAMED          CHAR (8)
            2 SEG_LEVEL          CHAR (2),
            2 STATUS_CODE        CHAR (2),
            2 PROC_OPTIONS       CHAR (4),
            RESERVE_DLI          FIXED BIN (31,0),
            2 SEG_NAME_FB        CHAR (8)
            2 LENGTH_FB_KEY      FIXED BIN (31,0),
            2 NUMB_SENS_SEGS     FIXED BIN(31,0)
            2 KEY_FB_AREA        CHAR (N);
```


DL/I Call Parameters

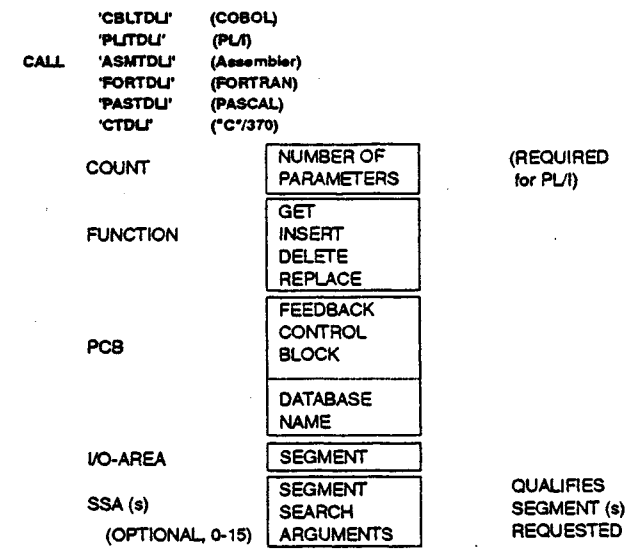


Figure 2-7. DL/I Call Parameters (CM011211)

Notes:

DL/I Function Codes

FUNCTION	CODE
GET SEGMENTS:	
GET UNIQUE	GUbb
GET NEXT	GNbb
GET NEXT WITHIN PARENT	GNPb
GET SEGMENT for DELETE or UPDATE:	
GET HOLD UNIQUE	GHUb
GET HOLD NEXT	GHNb
GET HOLD NEXT WITHIN PARENT	GHNP
DELETING:	
DELETE	DLET
UPDATING:	
REPLACE	REPL
ADDING:	
INSERT	ISRT

Figure 2-8. DL/I Function Codes (CM011213)

Notes:

CBLTDLI

```
CALL 'CBLTDLI' USING GN, EDPCB, COURSE-SEG.

FUNCTION CODE

"WHAT"?

ADDRESS OF PCB

"WHICH"?

PROGRAM WORKING-STORAGE

"WHERE"?


CALL PLITDLI (THREE, GU, EDPCB, COURSE_SEG);
```

Figure 2-9. CBLTDLI (CM011215)

Notes:

Identifying Required Segments

COURSE segment layout in EDUCATION database:

Course Code	Course Description	Course Length	Course Price
COURSCDE	COURSDES	COURSLEN	COURSPRC

A COURSE segment occurrence:

U3700	IMS Fundamentals	003	750.00

Figure 2-10. Identifying Required Segments (CM011217)

Notes:

The DBDGEN Implementation :

SEGM NAME = COURSE

FIELD NAME = (COURSCDE, SEQ, U) ...

FIELD NAME = COURSDES ...

FIELD NAME = COURSLEN...

FIELD NAME = COURSPRC ...

Segment Search Arguments (SSAs)

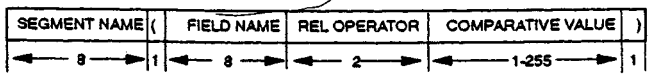
SSAs may be used to "qualify" a call in two ways:

1. Unqualified SSA to identify only the segment type

EXAMPLE: GET ALL COURSE SEGMENTS WITHIN A DATABASE

GN COURSEbb

2. Qualified SSA identifies specific occurrences of segments :



b= or EQ, ¬ = or NE
b> or GT, > = or GE
b< or LT, < = or LE

EXAMPLE: GET THE COURSE SEGMENT WITH KEY U3700

GU COURSEbb(COURSCDEb=U3700)

Figure 2-11. Segment Search Arguments (SSAs) (CM011219)

Notes:

COBOL Example:

```
01        SSA1.
            02 SEG_NAME        PICTURE X(8).
            02 BEGIN-QUAL      PICTURE X(1)    VALUE '('.
            02 FIELD-NAME      PICTURE X(8)
            02 REL-OPER        PICTURE X(2)
            02 SEARCH-VALUE    PICTURE X(N).
            02 END-QUAL        PICTURE X(1)    Value ')'.
```

PL/I Example:

```
DCL 1     SSA1                STATIC,
            2 SEG_NAME        CHAR (8),
            2 SEG_QUAL        CHAR (1),       INIT('(').
            2 SEG_FIELD_NAME  CHAR (8),
            2 SEG_REL_OPER    CHAR (2),
            2 SEG_KEY_VALUE   CHAR(n),
            2 SEG_END_QUAL    CHAR (1),       INIT(')');
```

DL/I Control-Block Relationships

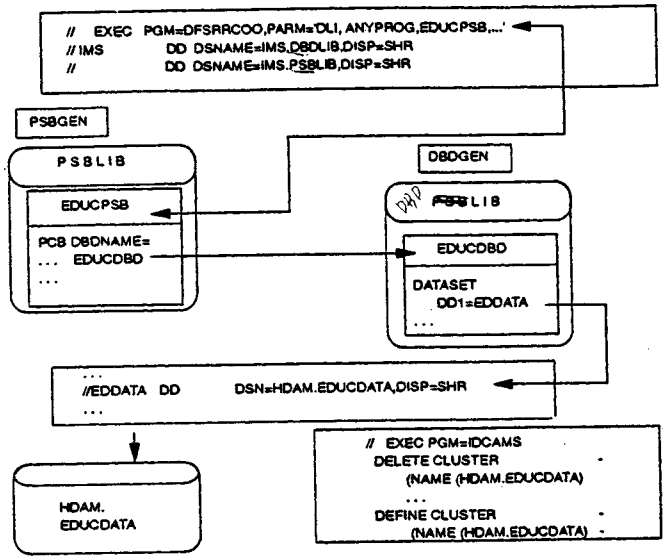


Figure 2-12. DL/I Control-Block Relationships (CM011221)

Notes:

Before the DL/I BATCH JOB called 'ANYPROG' can be executed:

1. The PSB called **EDUCPSB** must be in the PSB library
2. The DBD called **EDUCDBD** must be in the DBD library
3. The database data sets must exist

MVS Access Methods

TRANSFERRING DATA TO A PROGRAM:

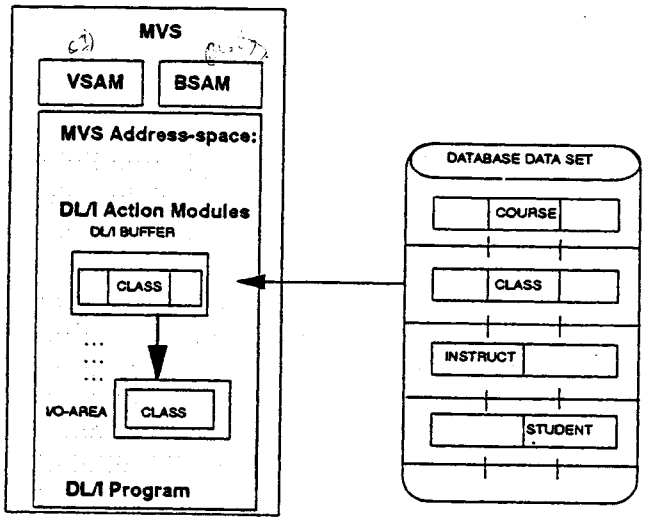


Figure 2-13. MVS Access Methods (CM011223)

Notes:

1. DL/I action modules formulate a request to VSAM or BSAM.
2. The unit of transfer from DASD to a buffer is a Control Interval (CI) in VSAM, or a block (BSAM)
3. The unit of transfer from DL/I to an application program is a segment

Retrieving Data

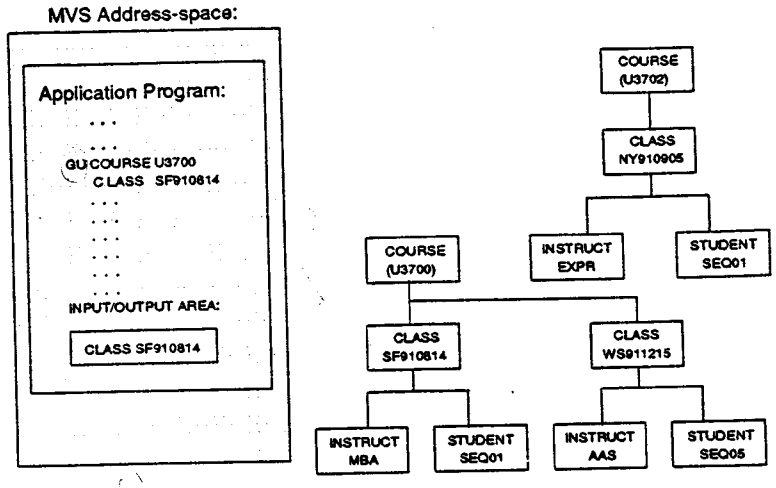


Figure 2-14. Retrieving Data (CM011225)

Notes:

Get Unique Calls

(GU and GHU)

- Used to retrieve a specific segment occurrence independent of the current position within the database
- Qualified SSAs identifying each hierarchic level are normally provided
- A GET-UNIQUE Call with missing levels of qualification uses current control block information for missing levels. If none exist, unqualified SSAs are assumed for the missing levels.
- A GET-UNIQUE call always returns the first segment in the data base that satisfies the qualification
- STATUS-CODES:

'GE' - Segment not found

'bb' - Successful CALL

Figure 2-15. GET UNIQUE Calls (CM011227)

Notes:

GE Segment NOT FOUND
AB No I/O Area Specified
AC Hierachical path error
AD Invalid Func Code
AJ SSA qualified format invalid
AK Invalid Field Name
AO I/O error

Database Record (DBR)

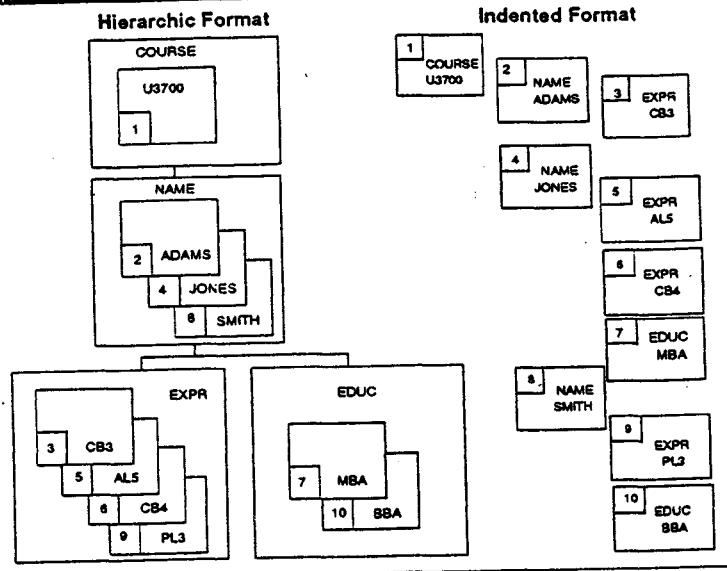


Figure 2-16. Database Record (DBR) (CM011229)

Notes:

GET UNIQUE Examples

Call	Cur Pos	SSA(s) in Call	Implied SSAs	Segm Retr	Stat Code
1		COURSEbb(CODE=U3700)		1	bb
2		COURSEbb(CODE=3700) NAMEbbbb(LASTNM=JONES)		4	bb
3A		COURSEbb(CODE=U3700) NAMEbbbb(LASTNM=JONES) EDUCbbbb(DEGREE=MBA)		7	bb
3B	7	EXPRbbbb(CLASSIF=AL5)	COURSE(U3700) NAME(JONES)	5	bb
3C	5	NAMEbbbb EXPRbbbb(CLASSIF=CB3)	COURSE(U3700)	3	bb
4A		None		1	bb
4B	1	EXPRbbbb(CLASSIF=CB4)	COURSE(U3700)	6	bb
4C	6	EXPRbbbb(CLASSIF=PL3)	COURSE(U3700) NAME(JONES)	-	GE
4D	6	COURSEbb(CODE=U3700) EDUCbbbb(DEGREE=MBA)	NAME(JONES)	7	bb

Figure 2-17. GET UNIQUE Examples (CM011231)

Notes:

GET NEXT Calls

(GN and GHN)

- Used to sequentially process segments in a database, or start sequential processing on a portion of it
- Always proceeds forward from current position within the database
- A GN Call with no SSAs returns the next segment in the database
- A GN Call with an unqualified SSA returns the next occurrence of that segment type
- A GN Call with a qualified SSA returns the next occurrence of the specified segment type which meets the search criteria
- Unqualified SSAs are supplied for missing levels
- STATUS-CODES:
  Unqualified Calls Only:
  'GA' - Crossed hierarchic boundary to higher level
  'GK' - Different segment type at the same level

  'GB' - End of Database reached
  'GE' - Segment not found
  'bb' - Successful call

Figure 2-18. GET NEXT Calls (CM011233)

Notes:

GET NEXT WITHIN PARENT

(GNP or GHNP)

WORKS JUST LIKE GN EXCEPT ONLY UNDER ESTABLISHED PARENT.

PARENT SET BY SUCCESSFUL GN or GU CALL.

STATUS CODES - SAME AS GN.

Figure 2-19. GET NEXT Within Parent (CM011235)

Notes:

GET NEXT Examples

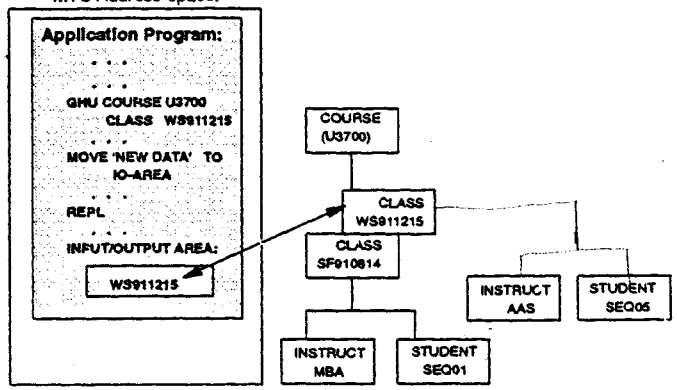
Call	Current Position	SSA(s) in call	Segm Retr	Status Code
1A			1	bb
B	1		2	bb
C	2		3	bb
D	3		4	GA
E	4		5	bb
F	5		6	bb
G	6		7	GK
H	7		8	GA
I	8		9	bb
J	9		10	GK
K	10		-	GB
L	(WRAP)		1	bb
2A		NAME	2	bb
B	2	NAME	4	bb
C	4	NAME	8	bb
D	8	NAME	-	bb
3A	1	NAMEbbbb(LASTNM=SMITH)	8	bb
B	8	NAMEbbbb(LASTNM=JONES)	-	GB
4	1	COURSEbb(CODE=U3700) NAMEbbbb(LASTNM=JONES) EXPRbbbb(CLASSIF=PL5)	-	GE

Figure 2-20. Get Next Examples (CM011237)

Notes:

Replacing Data

- A GET HOLD call ('GHx') must precede a 'REPL' call
-



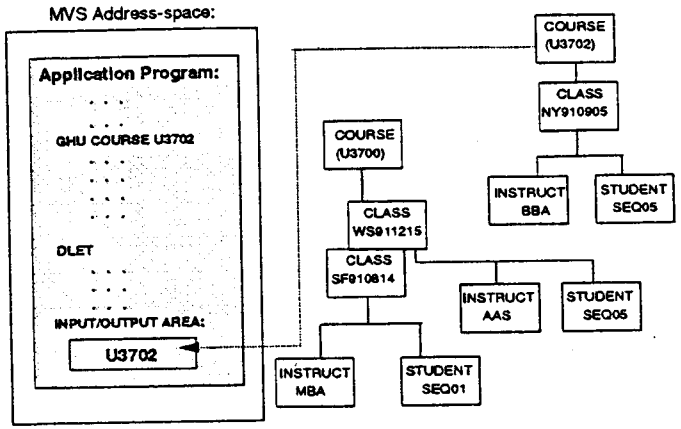
1. CLASS (WS911215 SEGMENT MOVED TO I/O-AREA
2. CHANGES ARE MADE TO DATA IN I/O-AREA
3. UPDATED SEGMENT REPLACED IN DATABASE

Figure 2-21. Replacing Data (CM011239)

Notes:

Deleting Data

- A *GET HOLD* call ('GHx') must always precede a 'DLET' call



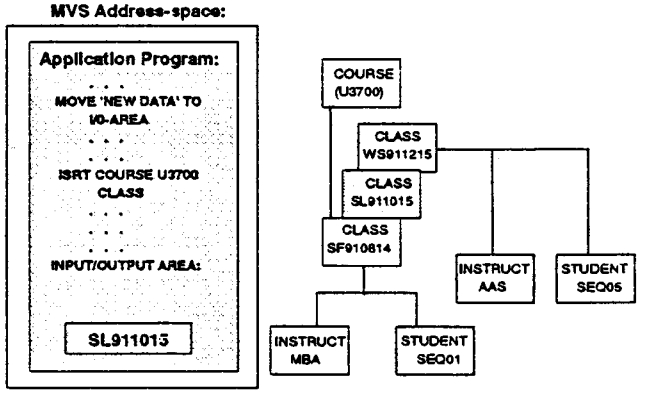
1. COURSE (U3702) SEGMENT MOVED TO I/O-AREA
2. SEGMENT AND ITS DEPENDENTS ARE ELIMINATED
3. WHAT IS IN DATABASE AFTER DELETION?

Figure 2-22. Deleting Data (CM011241)

Notes:

Inserting Data

- A POSITION IS ESTABLISHED FOR non-KEYED SEGMENTS
- KEYED SEGMENTS DO NOT REQUIRE PRIOR GET CALL



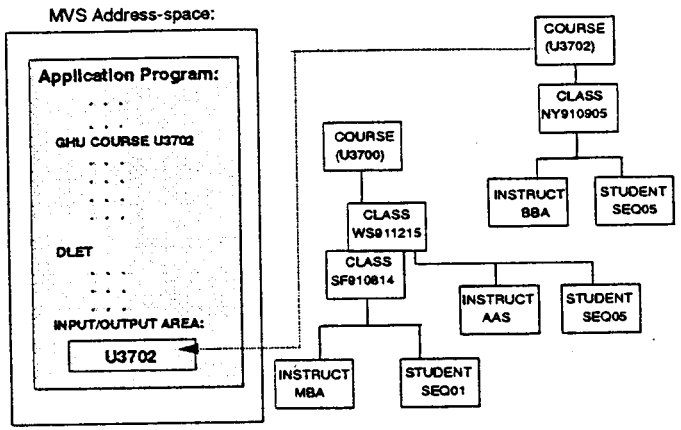
1. NEW SEGMENT DATA BUILT IN I/O-AREA
2. CLASS (SL911015) SEGMENT ADDED TO DATABASE

Figure 2-23. Inserting Data (CM011243)

Notes:

Deleting Data

- A *GET HOLD* call ('GHx') must always precede a 'DLET' call



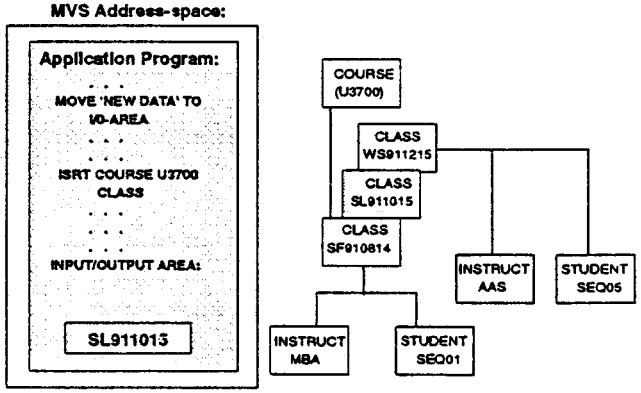
1. COURSE (U3702) SEGMENT MOVED TO I/O-AREA
2. SEGMENT AND ITS DEPENDENTS ARE ELIMINATED
3. WHAT IS IN DATABASE AFTER DELETION?

Figure 2-22. Deleting Data (CM011241)

Notes:

Inserting Data

- A POSITION IS ESTABLISHED FOR non-KEYED SEGMENTS
- KEYED SEGMENTS DO NOT REQUIRE PRIOR GET CALL



1. NEW SEGMENT DATA BUILT IN I/O-AREA
2. CLASS (SL911015) SEGMENT ADDED TO DATABASE

Figure 2-23. Inserting Data (CM011243)

Notes:

DL/I Test Program

Module 'DFSDDLT0' is provided:

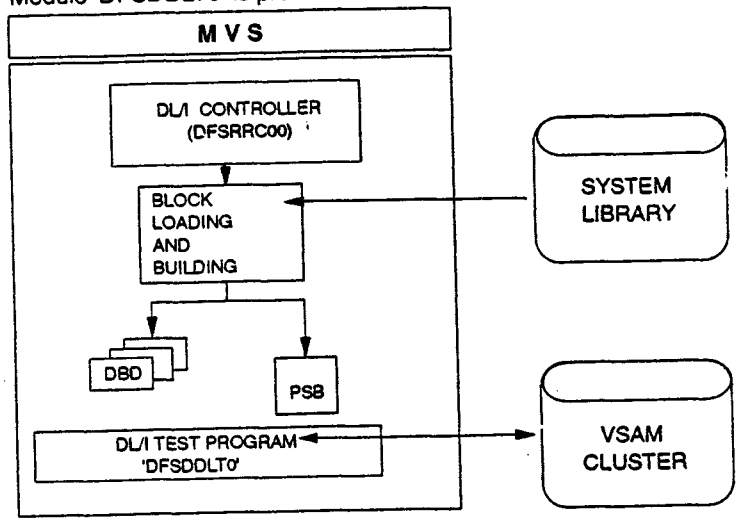


Figure 2-24. DL/I Test Program (CM011245)

Notes:

DL/I Test Program...

```
//DLT0      EXEC PGM=DFSRRC00,PARM 'DLI,DFSDDLT0,PSBCOBOL,,,,,,,,,N,N,  .
//STEPLIB   DD DSN=IMS3I.RESLIB,DISP=SHR
//IMS       DD DSH=SFN.U37OO.PSBLIB,DISP=SHR
//           DD DSH=SFN.U37OO.DBDIB,DISP=SHR
//PRINTDD   DD SYSOUT=
//SYSPRINT  DD SYSOUT=
//SKILL     DD DSN=SFSV.U370-T16.SKILL,DISP=SHR
//SYSIN     DD X
S 1 1 1 1 1   SKILL  2
UXXXXXXXXXXXX  TEST STATEMENT 1 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
L   GU  SKILL    (SKCLASS = BU115  )
        NAME     (FULNAM = AKERSON, LAWRENCE R.                    X
    CONT      )                                                    X
EH  OK
L  0006 GHP
/X
.....
IMS  3.1  TEST PROGRAM OUTPUT  XX BEGIN TEST XX  TIME= 10.57.06.93    DATE= 91.081

0002 0F 0003 PCB SELECTED =              DBD SELECTED = SKILL
.....
CALL=GU      SEG=SKILL      FIELD=SKCLASS  OPER=     VALUE=BU115     )
             SEG=NAME       FIELD=FULNAM   OPER=     VALUE=AKERSON, LAWRENCE R.

SEGMENT   = (AKERSON, LAWRENCE R.
             (                )
COMP                        RET CODE=OK
DBPCB   LEV=02 SEG=NAME     RET CODE=      KFDB LEN=05. KEY FDB=BU115  AKERSON, LAWRENCE R.
COMMENTS                                                    TIME=1-.57.09.49 DATE=91.081
.....
CALL=GNP

SEGMENT   = (05A100-30436     1000)

COMP                        RET CODE=OK
DBPCB   LEV=03 SE6=EXPR     RET CODE=      KFDB LEN=0054
  KEY FDBK= (BU115   AKERSON, LAWRENCE R.                        1000)

CALL=GNP

SEGMENT     = (06A200-10242   2000)

COMP                              RET CODE=OK
DBPCB   LEV=03 SE6=EXPR           RET CODE=      KFDB LEN=0054
 KEY FDBK=    (BU115    AKERSON, LAWRENCE R.                     2000)
   ...
   ...

STATISTICS FOR DATA BASE CALLS:
                        GU    GN    GNP   GHU   GHN   GHNP  ISRT  DLET   REPL   TOTAL
  DL/I COUNTS BY CALL-   1     0     6     0     0     0     0     0      0      7

  TOTAL NUMBER OF CALLS =  7  TOTAL NUMBER EQUAL COMPARES =   0  TOTAL NUMBER UNEQUAL COMPARES =

  PSB= PSBCOBOL PCB=        DBD= SKILL    STOP TIME= 10.57.12.27 DATE= 91.081
```

Figure 2-25. DL/I Test Program ... (CM011247)

Summary

All access to database is through the DL/I interface

Programming Language Independent

Logging and Recovery not a programming responsibility

Figure 2-26. Summary (CM011246)

Notes:

Unit 3. IMS Hierarchic Access Methods

What This Unit Is About

Various methods used by IMS to physically organize database information.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Identify the principal IMS access methods, their similarities, and their differences
- Discuss the application requirements which dictate the choice of an access method.

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 1 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8012-00 IMS/ESA V5 Administration Guide: DB

SC26-8034-00 IMS/ESA V5 Utilities Reference: DB

Objectives

- Identify the principal IMS access methods, their similarities, and their differences.
- Discuss the application requirements which dictate the choice of an access method.

Figure 3-1. Objectives (CM011301)

Notes:

3.1 IMS Hierarchic Access Method

IMS Hierarchic Access Methods

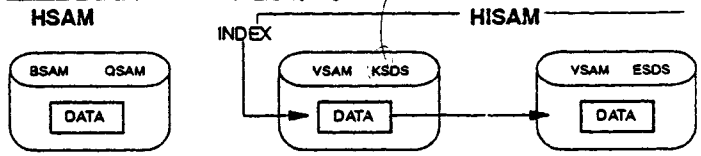
- Basic: Access Method is Transparent to Program
  - H SAM: Hierarchic Sequential
  - HISAM: Hierarchic Indexed Sequential
  - H DAM: Hierarchic Direct
  - HIDAM: Hierarchic Indexed Direct
- Specialized: MVS-compatible
  - SH SAM: Simple Hierarchic Sequential
  - SHISAM: Simple Hierarchic Indexed Sequential
  - G SAM: Generalized Sequential
- Fast Path:
  - DEDB: Data Entry Database

Figure 3-2. IMS Hierarchic Access Methods (CM011300)

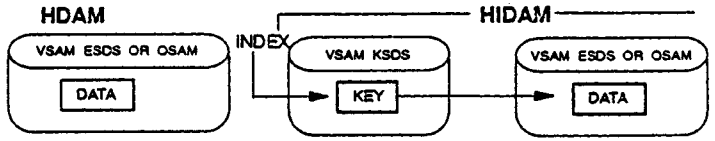
Notes:

Physical Storage

- HIERARCHIC SEQUENTIAL (HS):



- HIERARCHIC DIRECT (HD):



- SEQUENTIAL:

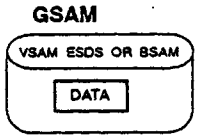
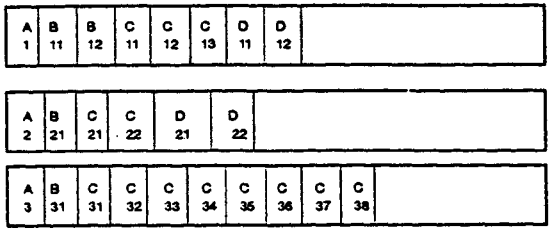


Figure 3-3. Physical Storage (CM011303)

Notes:

Physical Storage ...

- Sequential - Segments are related by physical adjacency



- Direct - Segments are related by pointers

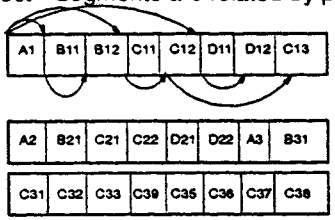


Figure 3-4. Physical Storage ... (CM011305)

Notes:

HD Pointers

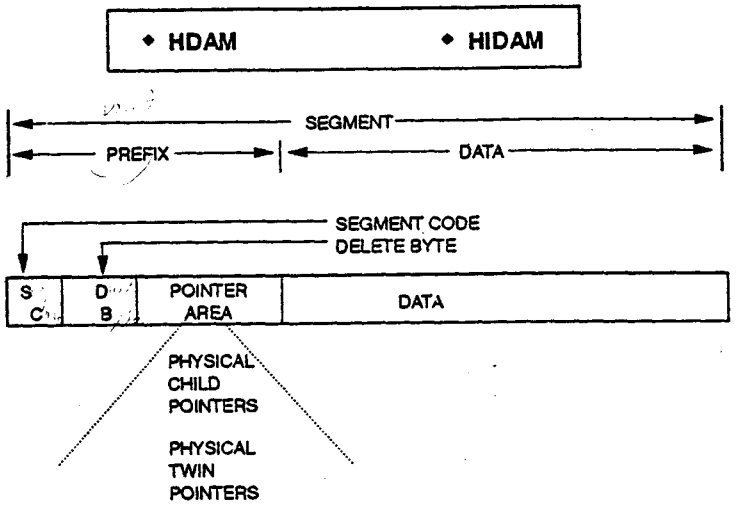


Figure 3-5. HD Pointers (CM011307)

Notes:

Hierarchic Database Records

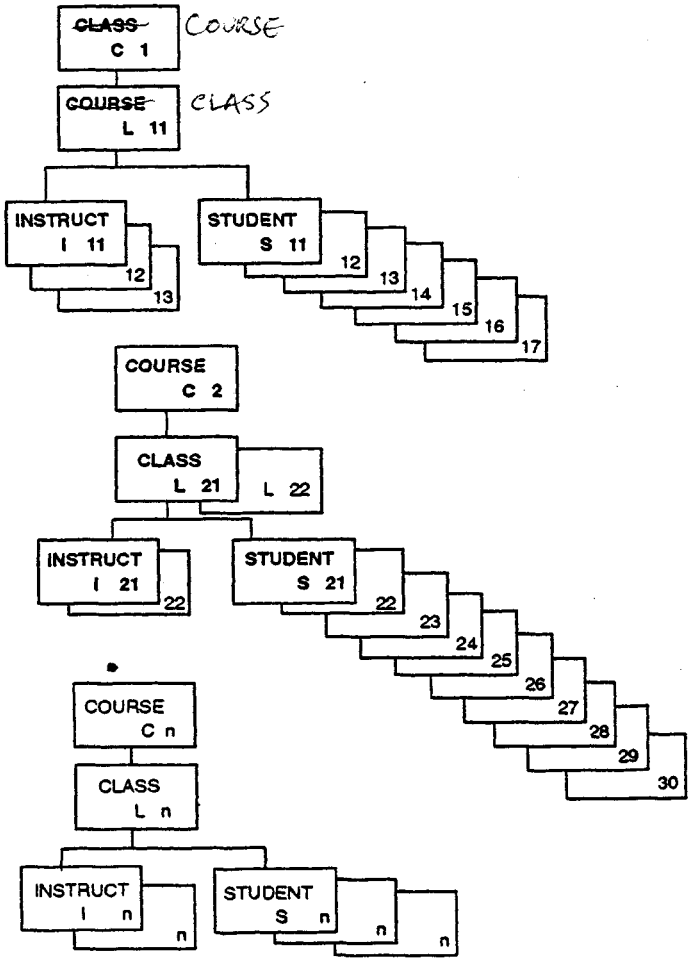


Figure 3-6. Hierarchic Database Records (CM011309)

Notes:

HSAM Storage Format

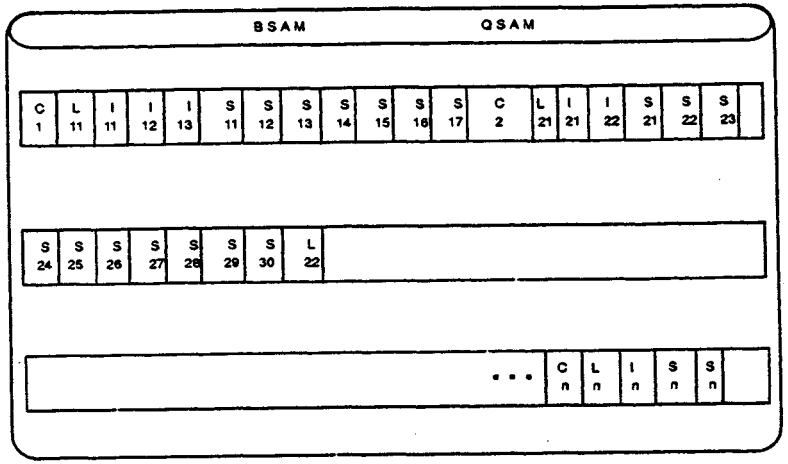


Figure 3-7. HSAM Storage Format (CM011311)

Notes:

1. There is no Delete or Replace. Update is via rewrite ('old master in, new master out').
2. HSAM is practical for historical data and archives that do not change much.

HISAM Storage Format

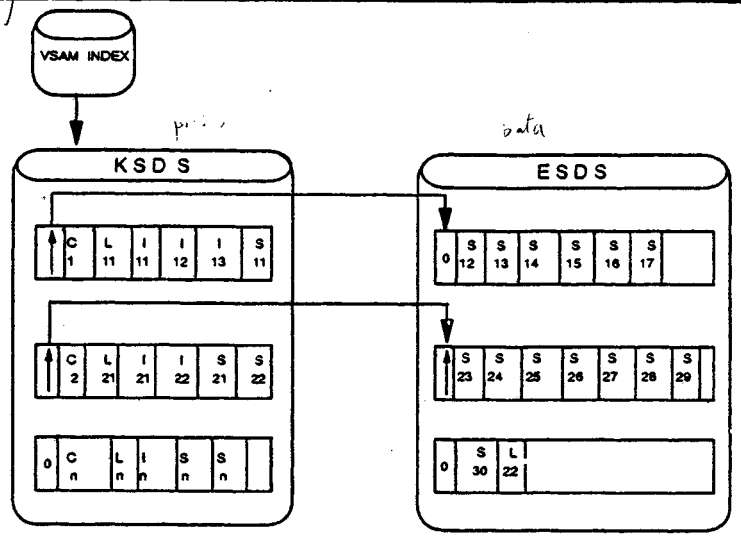


Figure 3-8. HISAM Storage Format (CM011313)

Notes:

HISAM is implemented only with a VSAM Key Sequenced Data Set (KSDS), and a VSAM Entry Sequenced Data SET (ESDS).

There may be one or more VSAM logical records per VSAM CONTROL INTERVAL

Unique root keys are required.

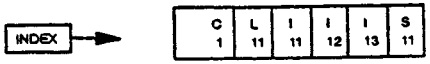
Segments for a database record are stored physically adjacent to one another.

In HISAM, VSAM records are "owned" by database records.

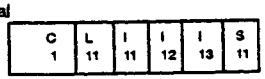
Storage space is not reusable in HISAM.

HISAM Summary

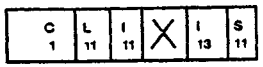
- Direct access to a database record is via a root key index built and maintained by VSAM



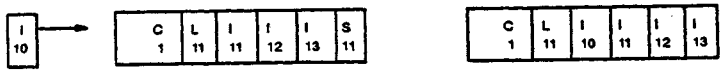
- Segments of a database record are related by physical adjacency



- Space of deleted segments is not reusable until reorganization



- Inserts require movement of data



- Numerous inserts and deletes can degrade performance

Figure 3-9. HISAM Summary (CM011315)

Notes:

1. Uses: Application that process data sequentially or 'skip-sequentially'
   - Through database root segments only, or
   - Through entire database records
2. Good for random root processing

HDAM Storage Format

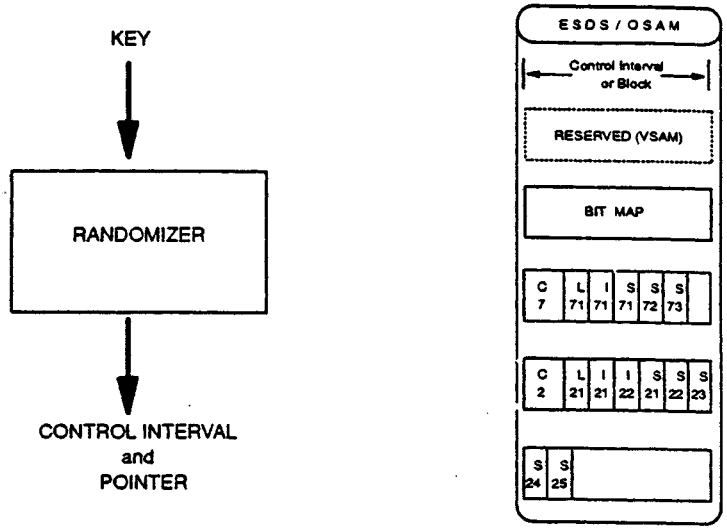


Figure 3-10. HDAM Storage Format (CM011317)

Notes:

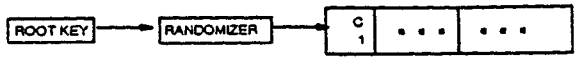
In HDAM, the data can be stored in a VSAM Entry Sequenced Data Set (ESDS) or an OSAM Data Set.

- If VSAM ESDS, the first CONTROLINTERVAL is Reserved
- IF OSAM, Use BLOCKs instead of CONTROLINTERVALS (CIs). (The first block is the BITMAP).

For all practical purposes, a VSAM ESDS and an OSAM Data Set are interchangeable in IMS.

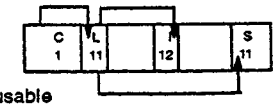
HDAM Summary

- Access to a database record is through a root segment via a randomizing module

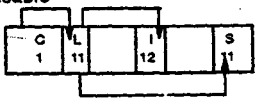


- Segments of a database record related by direct pointers

- Segments, once stored, never move



- Space of deleted segments is immediately reusable



- Uses: Primarily direct application processing requiring fast access to the root

Figure 3-11. HDAM Summary (CM011319)

Notes:

HIDAM Storage Format

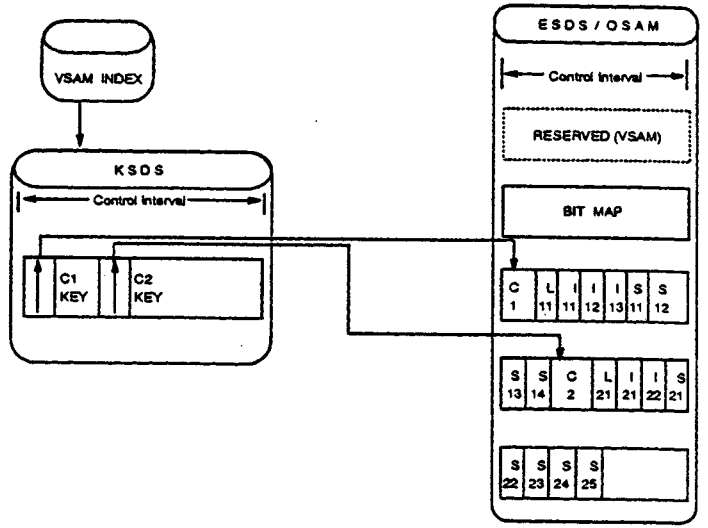


Figure 3-12. HIDAM Storage Format (CM011321)

Notes:

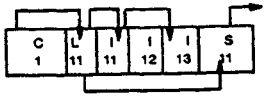
1. In HIDAM, only the key is stored in a VSAM Key-Sequenced Data Set (KSDS).
2. The data can be stored in a VSAM Entry-Sequenced Data Set (ESDS) or an OSAM Data Set.
   a. If VSAM ESDS, the first CI is Reserved.
   b. If OSAM, Use BLOCKs instead of Control Intervals (CIs). (The first block is the BITMAP).

For all practical purposes, a VSAM ESDS and an OSAM DS are interchangeable in IMS.

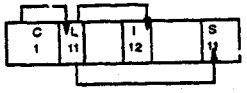
HIDAM Summary

- Direct access to a database record is via:
  - a root key index built and maintained by VSAM
  - a direct pointer maintained by IMS

- Segments of a database record related by direct pointers



- Segments, once stored, never move



- Space of deleted segments is immediately resuable

- Uses: Applications with both sequential and direct needs
  - large numbers of segments within the database record
  - large volume of insert and/or delete activity

Figure 3-13. HIDAM Summary (CM011323)

Notes:

Hierarchic Summary

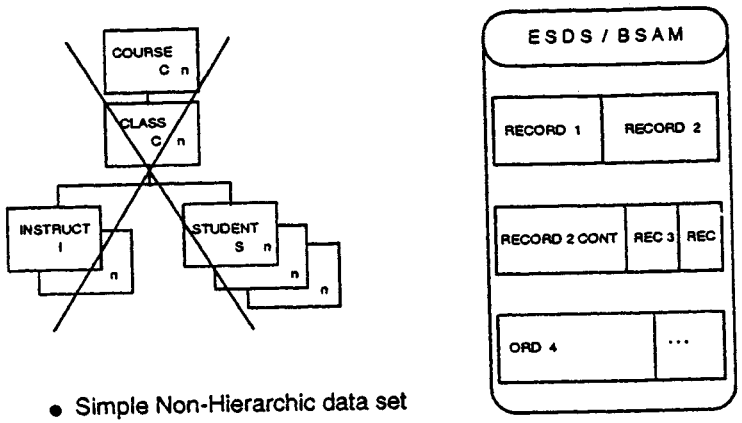
WHICH IMS ACCESS METHOD ?

- Identify the type of processing for each database:
  - Direct
  - Sequential
  - Both
- The "volatility" of the data
- The "variability" of the database record length

Figure 3-14. Hierarchic Summary (CM011325)

Notes:

GSAM



- Simple Non-Hierarchic data set
- Used mostly to ease restart process - more tomorrow.

Figure 3-15. GSAM (CM011327)

Notes:

- GSAM is implemented with MVS access methods BSAM or VSAM ESDS.
- Sequentially organized
- Fixed or variable length records
- No prefix
- Write checkpoint data set for subsequent file repositioning
- Available in IMS Batch or Batch Message Processing (BMP) environments
- Input data into programs, or extract data for reports.

Database Tools

Program Product

- DBD/PSB/ACB MAPPERs
- HD Pointer Checker
- HD Tuning Aid
- DB Segment Restructure
- HDAM Physical Sequence Sort/Reload (PSSR)
- Sequential Subset Randomizer
- Fast Reorganization Reload
- High Speed Sequential Retieval (HSSR)

Figure 3-16. Database Tools (CM011329)

Notes:

Database Tools

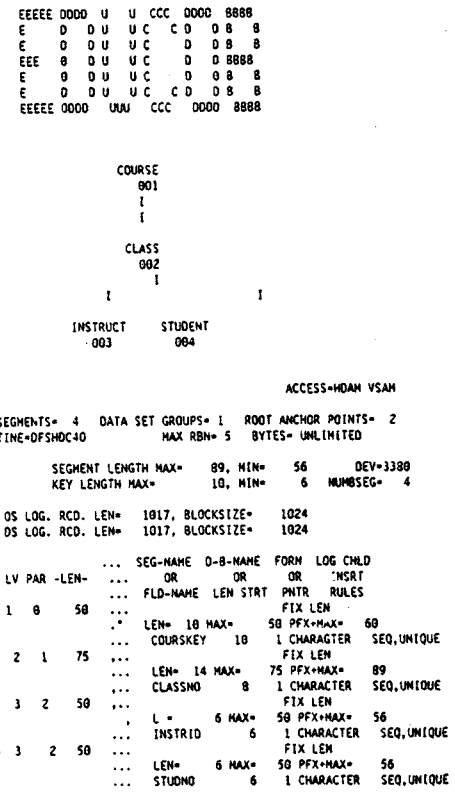


Figure 3-17. Database Tools (CM011331)

Notes:

Summary

HDAM for Direct Processing

HIDAM for mix of Sequential and direct processing

HD database segments linked by pointers, sophisticated space management

Figure 3-18. Summary (CM011332)

Notes:

Unit 4. IMS Other DL/I Functions

What This Unit Is About

Various optional features of IMS/DB.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss the implementation of logical relationships and reasons for their use
- Discuss the implementation of secondary indices and reasons for their use
- Discuss the implementation of variable length segments and reasons for their use
- Discuss the implementation of field level sensitivity and reasons for its use.

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 1 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8012-00 IMS/ESA V5 Administration Guide: DB

SC26-8034-00 IMS/ESA V5 Utilities Reference: DB

Objectives

- Discuss the implementation of logical relationships and reasons for their use.
- Discuss the implementation of secondary indices and reasons for their use.
- Discuss the implementation of variable length segments and reasons for their use.
- Discuss the implementation of field level sensitivity and reasons for its use.

Figure 4-1. Objectives (CM011401)

Notes:

4.1 IMS Other DL/I Functions

Other DL/I Functions

- LOGICAL RELATIONSHIPS
- SECONDARY INDEXING
- VARIABLE-LENGTH SEGMENTS
- SEGMENT EDIT/COMPRESSION
- FIELD-LEVEL SENSITIVITY

Figure 4-2. Other DI/I Functions (CM011400)

Notes:

Two Physical Databases

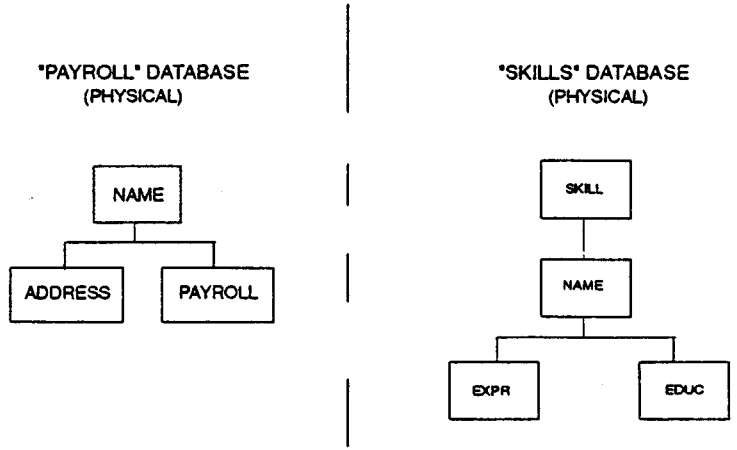


Figure 4-3. Two Physical Databases (CM011401)

Notes:

Two Physical Databases Cont.

2 PHYSICAL DATABASES

ARE MERGED IN A

PRODUCING A LOGICAL DBD

THE PCB (AND PROGRAM) USE THE NAME OF THE LOGICAL DATABASE

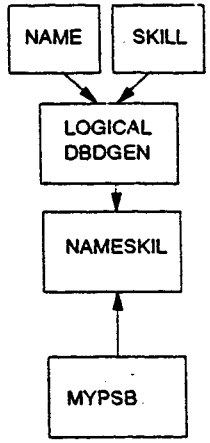


Figure 4-4. Two Physical Databases Continued (CM011403)

Notes:

Logical Database

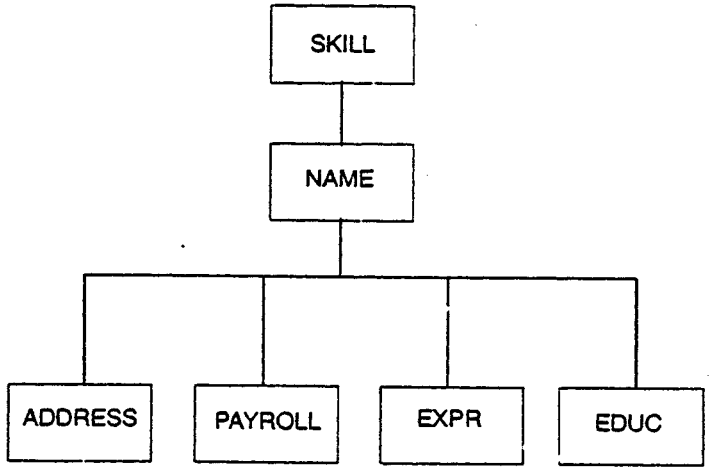


Figure 4-5. Logical Database (CM011405)

Notes:

Logical Database

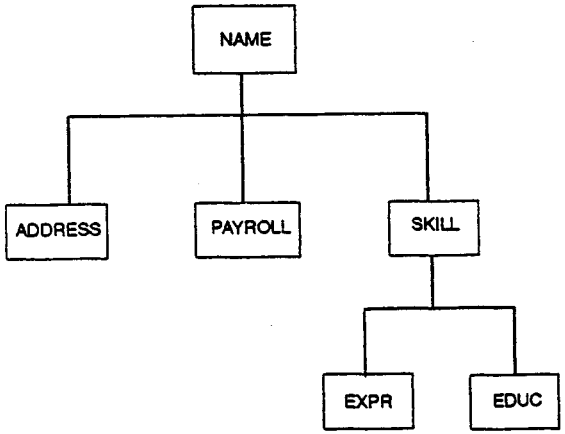


Figure 4-6. Logical Database(cont.) (CM011407)

Notes:

Two Views Of The Same Data

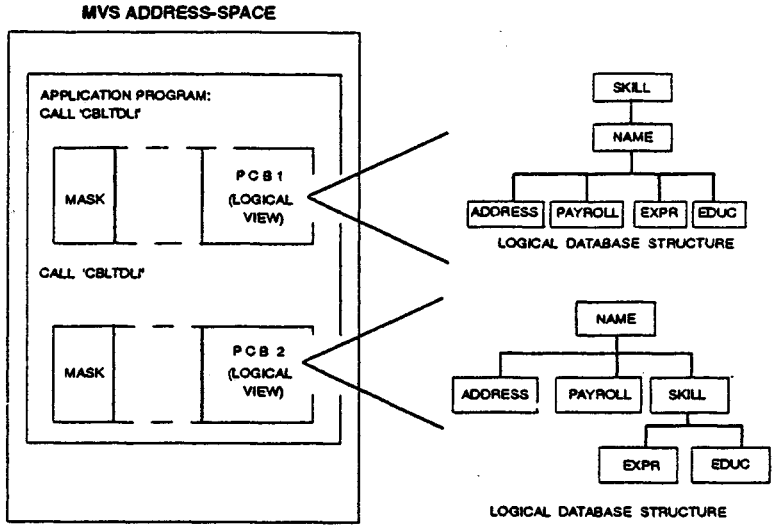


Figure 4-7. Two Views Of The Same Data (CM011417)

Notes:

IMS Secondary Indexing

- PROVIDE ALTERNATE SEQUENCE CAPABILITY
  - → PROVIDE ENTRY TO DATABASE ON VALUES OTHER THAN ROOT KEY
  - → PROVIDE MULTIPLE ENTRY VALUES INTO A COMMON DATABASE
- SECONDARY INDEXES ARE AUTOMATICALLY MAINTAINED BY THE SYSTEM

Figure 4-8. IMS Secondary Indexing (CM011419)

Notes:

IMS Secondary Indexing ...

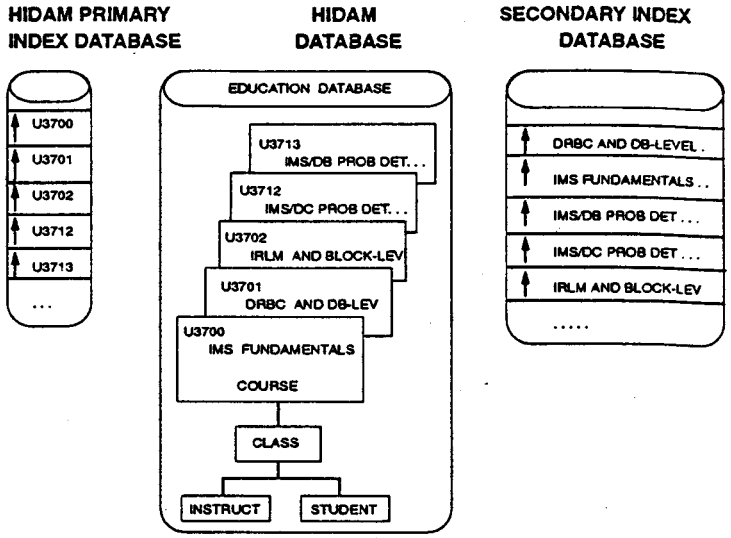


Figure 4-9. IMS Secondary Indexing... (CM011421)

Notes:

"Indexed " Database The HISAM, HIDAM, or HDAM database indexed into.

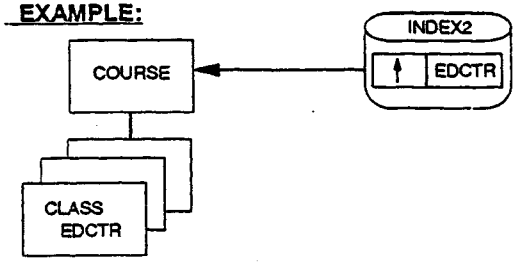
Pointer Segment The only type of segments in the secondary index database. The content of specified field(s) in the source segment is duplicated in the pointer segment.

Target Segment The segment that the application wants to retrieve. It is pointed to by the pointer segment. It can be the root or a dependent segment.

Source Segment The segment containing the field or fields that will be used to build the pointer segment.

IMS Secondary Indexing ...

- **EXAMPLE:**



- **PCB TYPE=DB,DBNAME=EDUCDB,PROCSEQ=INDEX2...**

- **CALL SEQUENCE:**

GU COURSEbb(EDCTRbbbb=CH)
GNP
INDEXED field may be used in SSA for TARGET segment

Figure 4-10. IMS Secondary Indexing... (CM011423)

Notes:

IMS Secondary Indexing ...

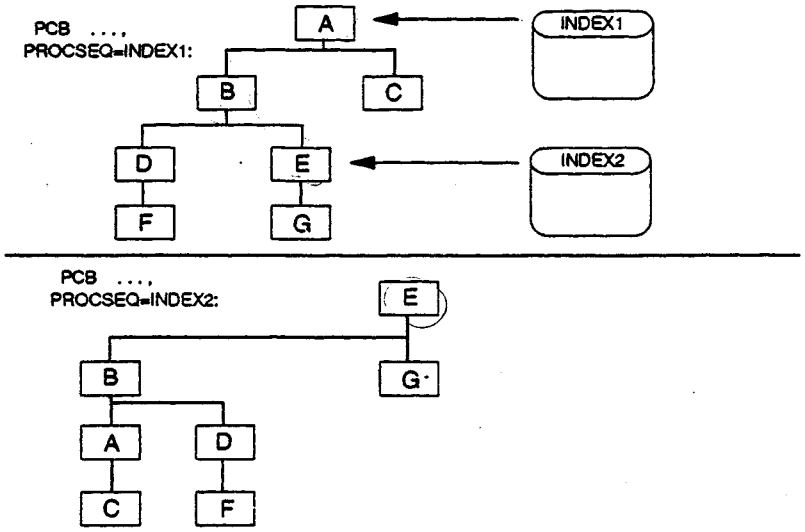


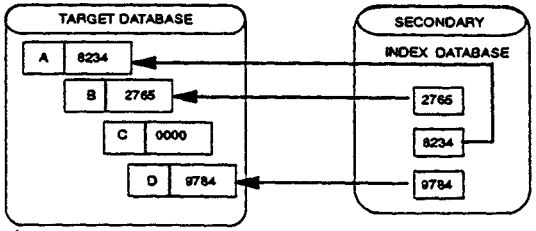
Figure 4-11. IMS Secondary Indexing... (CM011425)

Notes:

1. If the root is also the "target" segment, there is no inversion.
2. Optional use of multiple secondary indexes.
3. Restriction: can not ISRT or DLET E|B|A

Sparse Indexes

- A facility to suppress index entry creation for target segments
  - A common value
  - User-written



- Performance facility:
  - Reduces size of secondary index
  - Reduces maintenance

Figure 4-12. Sparse Indexes (CM011427)

Notes:

Seconding Indexing Summary

ADVANTAGES:

- SEQUENTIALLY PROCESS DATA IN A NON-KEY FIELD SEQUENCE
- RANDOM RETRIEVAL WHEN PRIMARY KEY IS NOT KNOWN
- SEQUENTIAL RETRIEVAL OF HDAM DATABASE
- EXCELLENT FOR PRODUCING REPORTS IN DIFFERENT SEQUENCES
- CHOICE OF EITHER PRIMARY OR SECONDARY INDEXES AS MAIN PROCESSING SEQUENCE

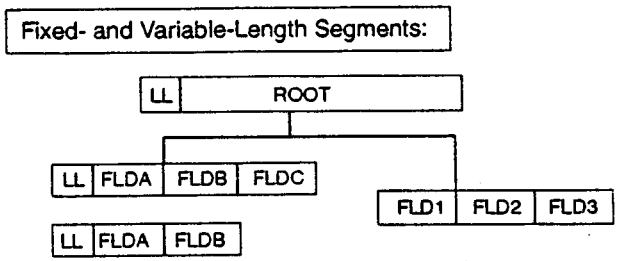
DISADVANTAGES:

- EXTRA PROCESSING AND I/O ASSOCIATED WITH THE MAINTENANCE OF SECONDARY INDEXES

Figure 4-13. Secondary Indexing Summary (CM011431)

Notes:

Variable-Length Segments



- FIXED-LENGTH SEGMENTS:
  - MULTIPLE OCCURENCES FOR VARIABILITY AND DASD BLOCKING
- VARIABLE-LENGTH SEGMENTS:
  - TEXT-PROCESSING APPLICATIONS
  - MULTIPLE OCCURENCES OF SAME FIELD, OR ONE VARIABLE-LENGTH FIELD (e.g. "Description")
  - "LL"-FIELD MUST BE MAINTAINED BY THE USER

Figure 4-14. Variable-Length Segments (CM011433)

Notes:

Variable-length segments are defined in the DBDGEN process:

```
SEGM NAME=xyz,BYTES=(max,min)
```


Segment Edit/Compression

OBJECTIVES:

- USER-WRITTEN ROUTINE RECEIVES CONTROL EACH TIME A SEGMENT IS ACCESSED:
  - TAKE ADVANTAGE OF VARIABLE-LENGTH SEGMENT FACILITY
  - IMPROVED SPACE UTILIZATION
  - PROVIDES ABILITY TO COMPRESS OUT REDUNDANT CHARACTERS
  - PROVIDES AN ENTIRELY NEW FACILITY FOR:
    1. SEGMENT EDIT
    2. ENCRYPTION/DECRYPTION FOR SECURITY
- ALLOW FUNCTIONS FOR FIXED- OR VARIABLE-LENGTH SEGMENTS

Figure 4-15. Segment Edit/Compression (CM011435)

Notes:

The Segment Edit/Compression exit is defined in the DBDGEN process:

```
SEGM NAME=xyz,COMPRTN=(routinename)
```

Segment Edit/Compression ...

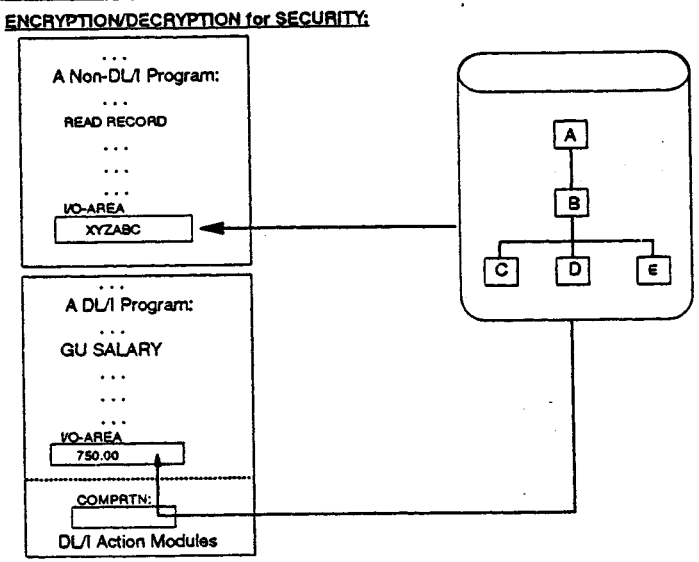


Figure 4-16. Segment Edit/Compression... (CM011437)

Notes:

Field-Level Sensitivity

EMPLOYEE segment in EMPLDATA database:

EMPNO	EMPNAME	ADDRESS	BIRTH-DATE	SALARY

VIEW-1 of the segment:

PCB TYPE=DB, DBNAME=EMPLDATA
SENSEG NAME=EMPLOYEE
SENFLD NAME=EMPNO
SENFLD NAME=EMPNAME
SENFLD NAME=ADDRESS

EMPNO	EMPNAME	ADDRESS

VIEW-2 of the segment:

PCB TYPE=DB, DBDNAME=EMPLDATA
SENSEG NAME=EMPLOYEE
SENFLD NAME=EMPNAME, REPL=YES
SENFLD NAME=BIRTHDATE, REPL=NO

EMPNAME	BIRTH-DATE

Figure 4-17. Field-Level Sensitivity (CM011439)

Notes:

Summary

Secondary indices allow multiple paths to data.

Logical relationships combines multiple databases into single program view.

Variable length segments with compression routines for efficient storage of text.

Figure 4-18. Summary (CM011440)

Notes:

Unit 5. IMS Transaction Manager

What This Unit Is About

How the IMS/TM software is used to manage access to an organization's information from a terminal network.

What You Should Be Able to Do

After completing this unit, you should be able to:

- List the principal components of the IMS Transaction Manager.
- List the different types of messages processed by IMS/TM
- Describe the flow of a transaction through the system.
- Discuss the security and integrity features of IMS/TM.

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 2 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8014-00 IMS/ESA V5 Administration Guide: TM

Summary

- List the principal components of the IMS Transaction Manager.
- List the different types of messages processed by IMS/TM.
- Describe the flow of a transaction through the system.
- Discuss the security and integrity features of IMS/TM.

Figure 5-1. Summary (CM011501)

Notes:

5.1 IMS Transaction Manager

Multiple IMS Batch Address-spaces

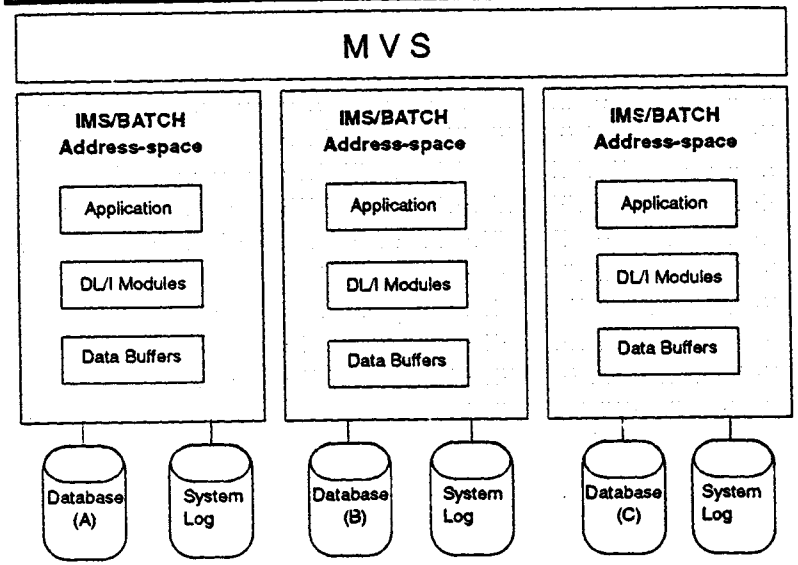


Figure 5-2. Multiple IMS Batch Address-space (CM011500)

Notes:

Goals Of The IMS/TM System

- High-volume, rapid-response system
- Transaction-Manager for application programs accessing:
  1. IMS/DB databases
  2. DB2 tables
- Growth capability
- Change independence
- Usage of common resources
- Programmer productivity
- Database management "integration"
- Data "integrity"
- Reliability/Availability/Serviceability (RAS) characteristics

Figure 5-3. Goals Of The IMS/TM System (CM011501)

Notes:

CICS → IMS/DB

IMS/TM (DC) → DB2

IMS/TM Architecture

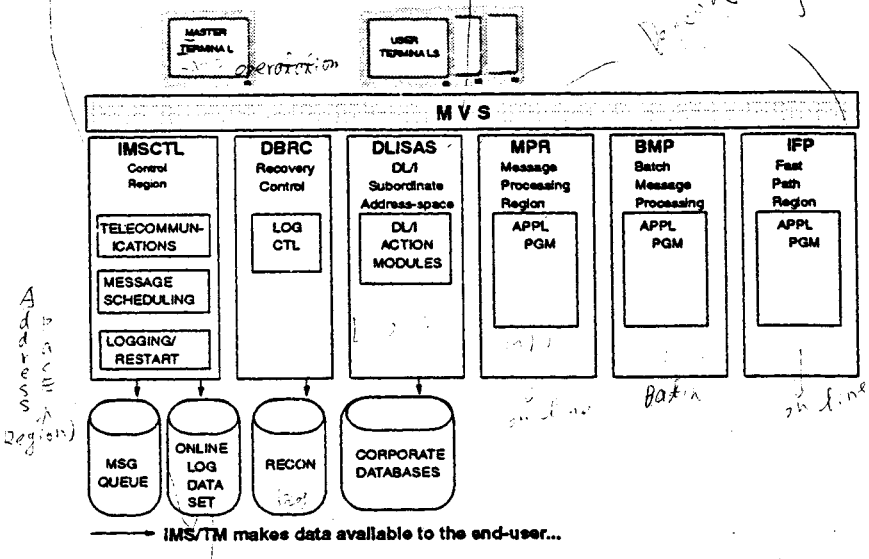


Figure 5-4. IMS/TM Architecture (CM011503)

Notes:

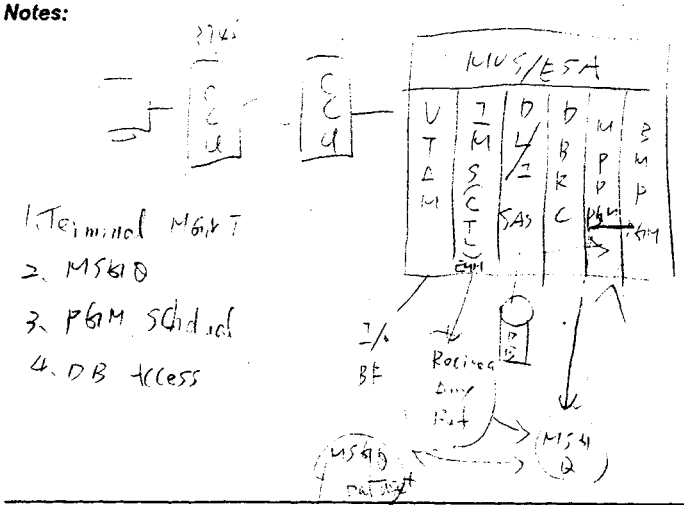
IMS/TM Dependent Regions

- Maximum of 1000 dependent regions

1. Message Processing Regions (MPRs):
   - Selective start-up at IMS initialization
   - Processes "online" transactions from terminals
   - IMSCTL automatically "schedules" application programs for execution to process IMS/DB databases, or DB2 tables
   - Many different application programs can process in a given MPR throughout processing period
2. Batch Message Processing (BMP) regions:
   - Started when JOB submitted by operations
   - Runs one application-program to process queued transactions, IMS/DB databases, and/or MVS files
   - Allows batch-type programs to process IMS queues and databases while IMS/TM system is available
3. Interactive Fast Path (IFP) regions:
   - Bypass IMS/TM scheduling via Expedited Message Handler (EMH)
   - For large transaction volumes that require fast response

Figure 5-5. IMS/TM Dependent Regions (CM011505)

Notes:



Logical Terminal (LTERM) Concept

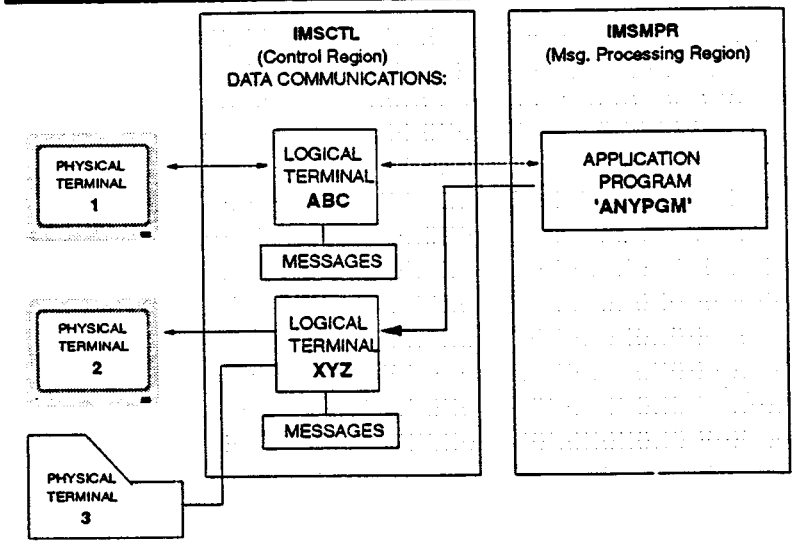


Figure 5-6. Logical Terminal (LTERM) Concept (CM011507)

Notes:

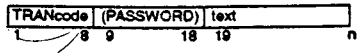
/ASS LTERM ... TO NODE ...

/ASS LTERM ... TO LINE ... PTERM ...

Message Types

- Three (3) types of messages:

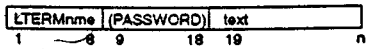
1. TRANSACTIONS:



Originate from terminal (most IMS/TM activity)

Can originate from another application-program via "program-switch"

2. MESSAGE-SWITCHES:



Allows terminals to communicate with each other

3. COMMANDS:

Mostly entered by MTO

Some commands may be entered from user terminals

Figure 5-7. Message Types (CM011509)

Notes:

Message Types ...

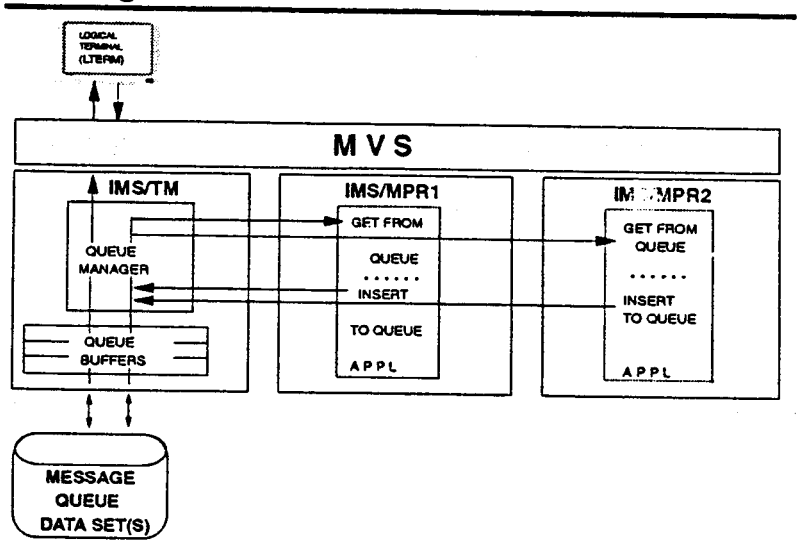
EXAMPLES:

1. **Transactions:**
   - **SALARY(XPSWD) 123456**
     ⟶ 'Salary for John Doe (123456) is $xxx.xx'
   - **ADDINV PART 9876 QTY=50**
     ⟶ 'Inventory on 9876: 10-Speed Bicycle = 9950 units'
2. **Message-Switches:**
   - **ACTGTERM**
     ACCOUNTING DEPARTMENT MEETING AT 2:00PM
3. **Commands:**
   - **/checkpoint freeze**
   - **/start line 5 pterm 2**
   - **/change(zzzyyx) password 999888 to passwo d 888999**

Figure 5-8. Message Types... (CM011511)

Notes:

Message Queue



- All INPUT and OUTPUT messages are queued...

Figure 5-9. Message Queue (CM011513)

Notes:

Message Queue ...

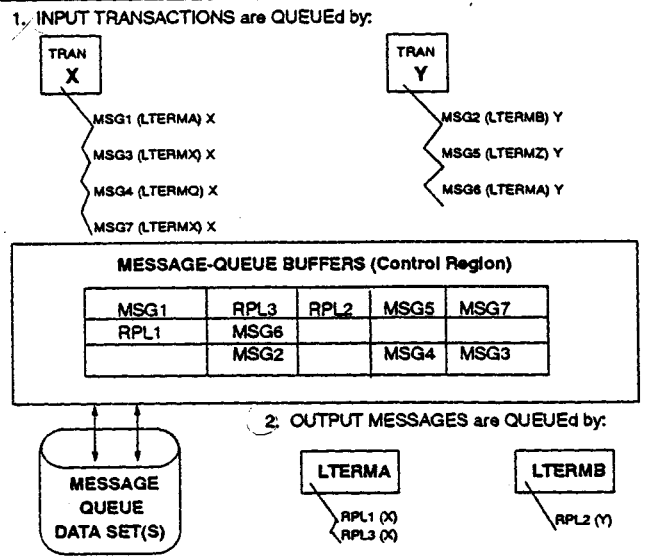


Figure 5-10. Message Queue... (CM011515)

Notes:

PSB Transaction Manager

PCB **TYPE=TP**

PCB **TYPE=DB**

PSBGEN...

TP **ALTERNATE RESPONSE IOPCB. USED TO SEND A MESSAGE.**

DB **USED TO READ OR UPDATE A DATABASE - SAME AS IMS DB.**

Figure 5-11. PSB Transaction Manager (CM011517)

Notes:

Logical Structures

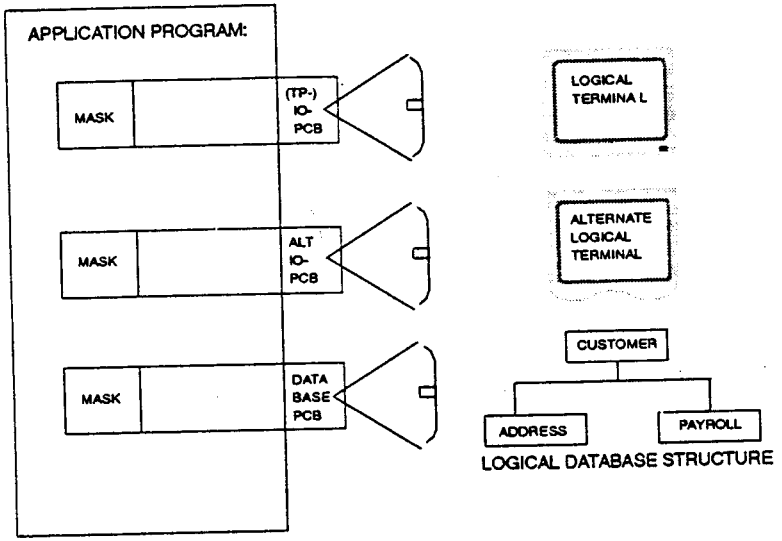


Figure 5-12. Logical Structures (CM011519)

Notes:

Message Format Service (MFS)

- MFS modules remove device- and line-control characters by reading pre-defined control blocks

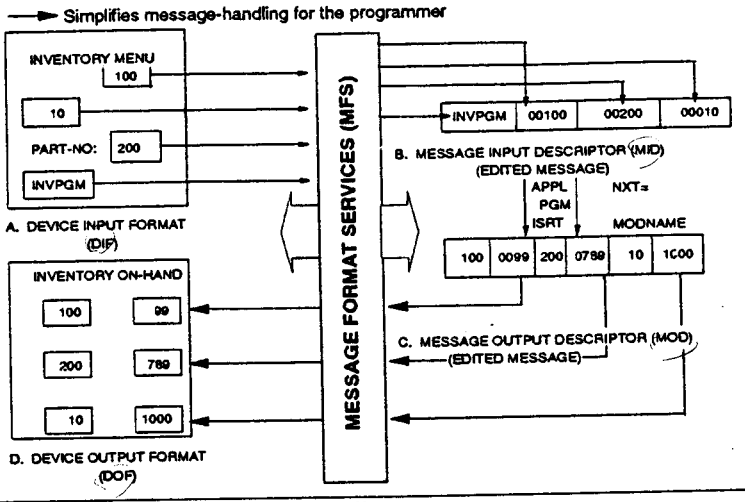


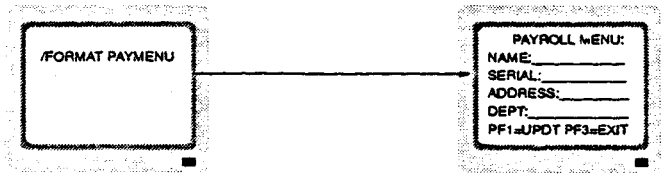
Figure 5-13. Message Format Service (MFS) (CM011521)

Notes:

Message Format Service

FORMATTING A SCREEN:

1. Using /FORMAT command:



——→ **No program is involved**

2. Using Program-control:

Format screen using output message from a program

CALL 'CBLTDLI' USING ISRT IO-PCB IO-AREA PAYMENU

——→ **Message that is written to queue contains MOD name to be used to map a DOF (screen-format) name**

Figure 5-14. Message Format Service (MFS) (CM011523)

Notes:

Message Format Service (MFS) ...

CREATING MFS CONTROL BLOCKS:

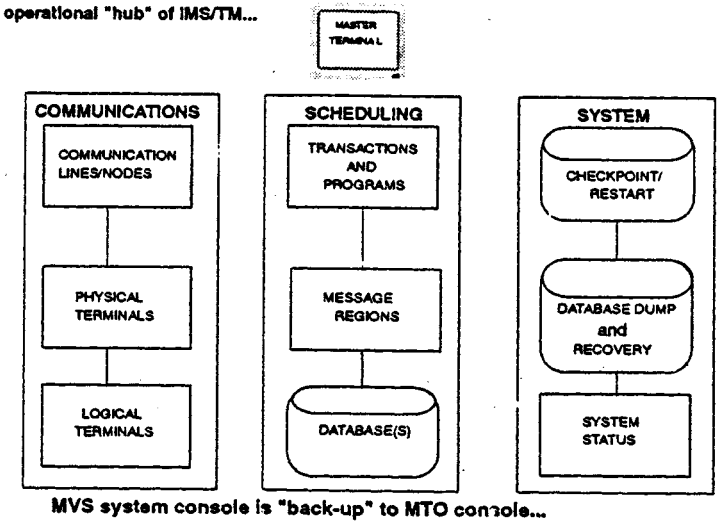
```
//TEST      EXEC    @MFS
//SYSIN       DD    *
      PRINT  NOGEN

SF00    MSG     TYPE=OUTPUT, SOR=(SFIL00, IGNORE), OPT=2,NXT=SFIN00,FILL=PT
        MFLD    REPLY, LTH=50,JUST=L
        MFLD    SKILL, LTH=8, JUST=L
        MFLD    NAME,LTH=6, JUST=L                    Creates MOD...
      MSGEND

SFIN00  MSG     TYPE=INPUT,SOR=SFIL00,OPT=2, NXT=SF00
        MFLD    TRANCD,LTH=6,FILL=NULL
        MFLD    TYPESEG,LTH=1,JUST=L, FILL=C' '
        MFLD    SNE, LTH=18,JUST=L, FILL=C' '
        MFLD    DEGREE, LTH=1, JUST=L, FILL=C' '
        MFLD    SUBJECT, LTH=15, JUST=L,FILL=C' '
        MFLD    CDATE,LTH=4, JUST=L,FILL=C' '         Creates MID...
      MSGEND

SFIL00  FMT
SFIL00  DEV TYPE=(3270,2) ,FEAT=IGNORE,
              PFK=(TRANCD,'CONVA '),
              DSCA=X'00A0'                          Creates DIF/DOF...
        DIV TYPE=INOUT
        DPAGE CURSOR=((5,7))
        DFLD 'SKILL INVENTORY UPDATE',POS=(2,30),ATTR=(NUM,PROT,HI)
        DFLD 'TYPE',POS=(5,2), ATTR=(NUM,PROT,HI)
TYPESEG DFLD    POS=(5,7), LTH=1
        DFLD 'SKILL/EMPLOYEE/COURSE',POS=(7,2),ATTR=(NUM,PROT,HI
SNE     DFLD    POS=(7,24),LTH=18
        DFLD 'DEGREE',POS=(9,2),ATTR=(NUM,PROT,HI)
DEGREE  DFLD    POS=(9,9),LTH=1
        DFLD 'SUBJECT',POS=(11,2),ATTR=(NUM,PROT,HI)
SUBJECT DFLD    POS=(9,9),LTH=1
        DFLD 'COMPLETION DATE',POS=(13,2),ATTR=( NUM,PROT,HI)
CDATE   DFLD    POS=(13,18),LTH=4
        DFLD 'CURRENT SKILL',POS=(15,2),ATTR=(NUM,PROT,HI)
SKILL   DFLD    POS=(15,16),LTH=8
        DFLD 'CURRENT NAME',POS=(15,25),ATTR=(NUM,PROT,HI)
NAME    DFLD    POS=(15,39),LTH=6
        DFLD 'RESULT',POS=(17,2),ATTR=(NUM,PROT,HI)
REPLY   DFLD    POS=(17,9),LTH=50
        DFLD 'USE PF KEY 1 TO BEGIN THE CONVERSATION',POS=(23,2)
      FMTEND
        END
```

Figure 5-15. Message Format Service(MFS) (CM011525)

Notes:

Master Terminal Operator (MTO)

A system resources controller LTERM that is the operational "hub" of IMS/TM...



MVS system console is "back-up" to MTO console...

Figure 5-16. Master Terminal Operator (MTO) (CM011527)

Notes:

Example Terminal Commands

MASTER Terminal:	REMOTE Terminal:
/NRESTART "Normal" /ERESTART "Emergency" /CHECKPOINT	
/DISPLAY /ASSIGN /CHANGE /STOP and /START NODE DATABASE PROGRAM TRANSACTION REGION	/SIGN ON USERID PAYROLL OFF /AM LTERM PAYROLL /LOCK LTERM PAYROLL
/BROADCAST NODE ALL PAYROLL DATABASE UNAVAILABLE UNTIL 2:00PM	/HOLD /RELEASE CONV 004 /EXIT
/RMLIST DBRC='DB DSD(DIVNT204) DBDS' /DBDUMP /DBRECOVERY	/FORMAT PAYMENU /TEST /EXCLUSIVE /RDISPLAY MASTER
/OPNDST /CLSDST /MODIFY PREPARE /MODIFY COMMIT	/RCLSDST

Figure 5-17. Example Terminal Commands (CM011529)

Notes:

IMS Security

- What Can be Secured?
  1. Terminals, Passwords, USERID Sign-on Verification
  2. Transactions
  3. Databases (Logical and Physical)
  4. Commands
- IMS Security Maintenance Utility (SMU) maintains a dual "matrix" for security:
  - ——→Protects IMS Resources/Objects
  - ——→IMS-provided Internal Security
- User-Written Exit Routines:
  1. /SIGN ON Exit Routine (DFSCSGN0)
  2. Transaction Authorization Exit Routine (DFSCTRN0)
  3. Security Reverification Exit Routine (DFSCTSE0)

Figure 5-18. IMS Security (CM011531)

Notes:

Terminal and Password Security

1. LTERM- Transaction Code matrix:
   ——→ Terminals can be restricted to designated applications:

	PAYING	REFIND	ADDINV	ADDPART	TABLEQ	DLETINV	VENDORD	DISBURS	REPLINV	SALARY	LISTINV	UPDTINV
REC101	1											
SAL102		1	1	1								
INV105			1		1	1	1		1	1	1	
INV106				1				1				1
INV107	1									1		

2. Password-Transaction Code matrix:
   ——→ Designated applications can be assigned a "password":

	PAYING	REFIND	ADDINV	ADDPART	TABLEQ	DLETINV	VENDORD	DISBURS	REPLINV	SALARY	LISTINV	UPDTINV
PSWD01	1											
PSWD02		1										
PSWD05			1									
PSWD06				1								

Figure 5-19. Terminal and Password Security (CM011533)

Notes:

SMU
IMSP - MATRIX
XA
+S

System Integrity Functions

- **Common Logging Facility:**
  ——→ **Database changes, transactions, system status changes**
- **Automatic system checkpoints**
  ——→ **Based on system activity**
  ——→ **System Recovery and Emergency Restart**
- **Database Recovery Control**
  ——→ **Similar to batch recovery operations** DBRC
- **Program Isolation** (PI)
  ——→ **Automatic backout for program ABENDs**
  ——→ **Data Sharing by multiple programs**
- **Normal Restart**
  ——→ **Start with unprocessed input/output from previous execution**

Figure 5-20. System Integrity Functions (CM011537)

Notes:

Common Logging Facility

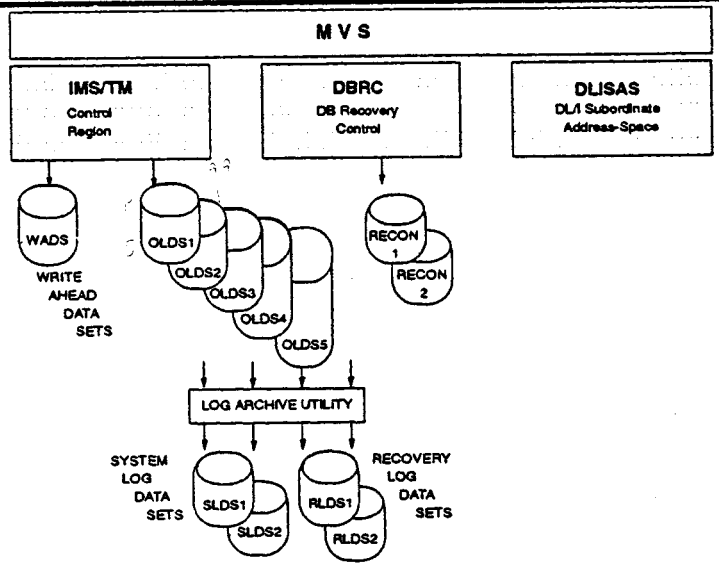


Figure 5-21. Common Logging Facility (CM011539)

Notes:

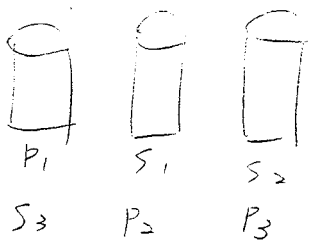
1. System Log Data Set (SLDS): May contain data from one or more OLDSs; used primarily for database recovery or Emergency Restart.
2. Recovery Log Data Set (RLDS): Contains only log records needed for database recovery.

Common Logging Facility

- Recording of IMS/DB and IMS/TM system activity:
  - "Before" and "After" images of changed DL/I segments
  - System status via periodic checkpoints
  - Input and Output messages are logged
- Provides integrity and recoverability of DB/TM systems:
  - Used for Dynamic Backout of failing transactions
  - Used for system "warm" and "emergency" restarts
  - Used for database recovery
  - Used for performance and accounting statistics
- Log Write-Ahead (LWA):
  - To guarantee database integrity, log records are physically written before database physically written
  - Only selected "events" qualify for this technique

Figure 5-22. Common Logging Facility... (CM011541)

Notes:



OLDS and WADS

- Online Log Data Set (OLDS):
  - Contains log records needed for:
    - Restart
    - Recovery
    - Backout: Batch and Dynamic
  - Consists of multiple data sets
  - Allows IMS/TM to continue logging when an "OLDS switch" occurs because of the following:
    - OLDS is full
    - I/O error while writing to an OLDS
    - Operator commands /DBR or /DBD
  - Written in a "wrap-around" manner
  - Can be archived automatically to the System Log Data Set (SLDS)
- Write Ahead Data Set (WADS):
  - Contains committed log records not yet written to OLDS
    - Normally, only full blocks are written
  - May be used to "close" the OLDS in an emergency

Figure 5-23. OLDS and WADs (CM011543)

Notes:

Database Recovery Control (DRBC)

THREE LEVELS OF CONTROL:

1. LOG-CONTROL, controller of IMS log data sets:
   - → Maintains information about OLDS status, e.g.
     ARCHIVE-NEEDED AVAILABLE ARCHIVE-IN-PROGRESS NEXT-OLDS
2. RECOVERY-CONTROL can assist in database recovery:
   - → Database "registration" by database data set
   - → Monitors Database Image Copies (ICs), Change Accumulations (CAs), Recoveries (DBRs)
   - → Automatic JCL generation of needed job-streams
3. SHARE-CONTROL enables database sharing:
   - → Access-intent must be registered with a "share-level":
     a) Database-level sharing
     b) Intra- or inter-processor block-level sharing

Figure 5-24. Database Recovery Control (DBRC) (CM011547)

Notes:

"The RECON" for Log-Control

- Recovery Control Data Sets:
  - Automatic recording of information about:
    - Online Log Data Set (OLDS)
    - System Log Data Set (SLDS)
    - Recovery Log Data Set (RLDS)
- Generation of utility JCL via DBRC command for:
  1. Log Archive
  2. Log Recovery

Figure 5-25. "The RECON" for Log-Control (CM011549)

Notes:

"The RECON" for Log-Control ...

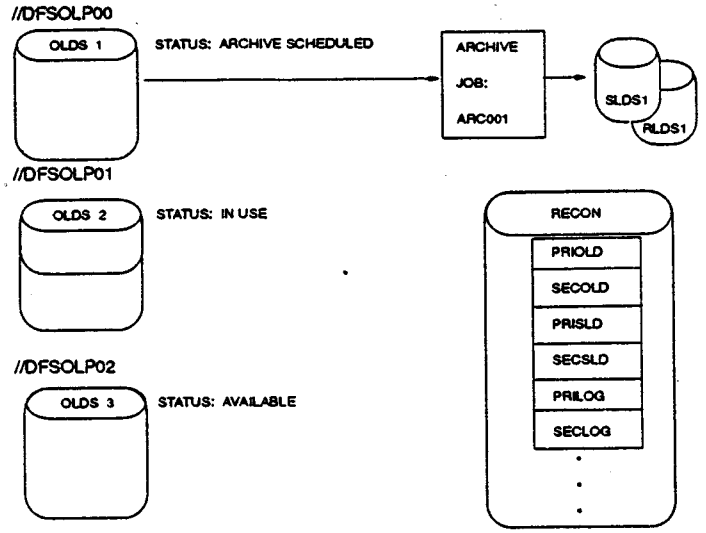


Figure 5-26. "The RECON" for Log-Control.... (CM011551)

Notes:

Information recorded about logs includes:

1. Subsystem ID
2. DDNAME
3. Data Set name
4. Start timestamp
5. Stop timestamp
6. Status
7. Archive timestamp
8. Archive JOB name

DBRC Database "Registration"

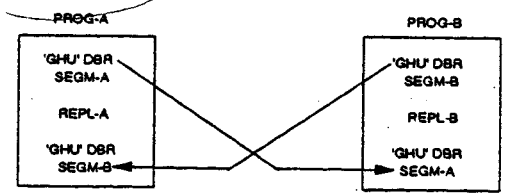
- Advantages:
  - Records recovery-related information in RECON
    - Example: which OLDS has log data for each database data set (DBDS) registered with DBRC
  - Controls recovery of databases
  - Generates JCL for recovery-related utilities
    - Used to select correct data sets to be used by Recovery and Image-Copy
    - Consolidates changes of log records
    - Saves time
    - Eliminates errors
- Register databases with DBRC commands

Figure 5-27. DBRC Database "Registration" (CM011553)

Notes:

Program Isolation

- Program "Isolation" (PI) facility:
  - Allows multiple online programs to process the same database for update while preserving integrity
  - → But not the same segment occurrence!
- Dynamic Backout facility:
  - Removes the effects of any programs that abnormally end (ABEND)
  - → Eliminates partial updates; uncommitted updates are backed-out
- Deadlock Detection:



PI elects "loser" transaction for "pseudo"-ABEND and dynamic back-out; re-schedules it later...

Figure 5-28. Program Isolation (CM011555)

Notes:

BMP
FP UTL
MPP
FP

Checkpoint and Restart

IMS uses its own CHECKPOINT processing:

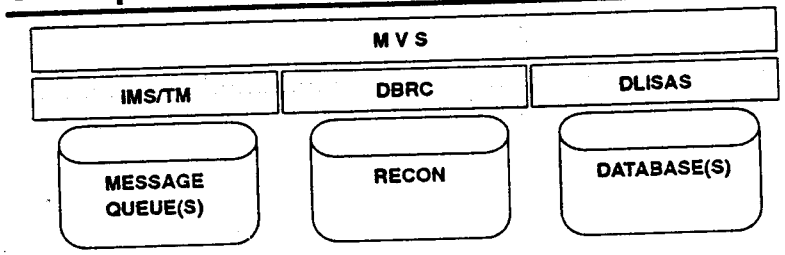
- System-wide checkpoint is taken periodically based on volume of log-records generated
- Restart is based on use of a CHECKPOINT which contains a "snapshot" of system control-blocks to log data set (OLDS)

Restart Type:	Checkpoint Selected:
COLD	CHKPT 0
WARM	CHKPT Generated At Previous Shutdown
EMERGENCY	Last CHKPT Written to Log

Figure 5-29. Checkpoint and Restart (CM011557)

Notes:

Checkpoint and Restart ...



1. MVS System Failure
2. Stand-alone dump
3. Restart MVS operating system
4. Emergency Restart IMS:
   - Database backout of "in-flight" transactions
   - "In-flight" transactions re-scheduled
   - Message queue reconstruction
   - Message transmisssion repositioning
5. Continue processing

Figure 5-30. Checkpoint and Restart... (CM011559)

Notes:

IMS Generation (IMSGEN)

- IMS is installed using an IMSGEN process (Macros define IMS resources):

Macro	Function
IMSCTRL	System-wide parameters
IMSCTF	MVS parameters
MSGQUEUE	Message Queue definitions
BUFPOOLS	Control-block buffer-pools
DATABASE	Databases
APPLCTN	Applications/Programs (PSBs)
TRANSACT	Transactions
COMM	Communications defaults
TERMINAL	Physical Terminals
NAME	Logical Terminals (LTERMs)

- Online Change facility permits DATABASE, APPLCTN, and and TRANSACT macro changes while "production" system is available
  - Also maintains MFS format blocks and security matrix
  - Implemented using a series of commands and utilities
  - Other changes require an IMSGEN
- ETO feature allows dynamic terminal network.

Figure 5-31. IMS Generation (IMSGEN) (CM011561.)

Notes:

Extended Terminal Option (ETO)

- Feature of IMS/TM
- Terminals may optionally be defined outside of the IMSGEN Process.
- Fewer IMSGEN's
- Smaller and faster IMSGEN's
- Dynamic control block management
- Requires user sign on

Figure 5-32. Extended Terminal Option (CM011563)

Notes:

IMS Related Products

- DB2 (External Subsystem DBMS) with IMS/TM
- CICS/ESA with IMS/DB ("local DL/I"), or
- CICS/ESA with IMS/DBCTL
- Screen Definition Facility II (SDF II)
  - → Help to build MFS control blocks
- Query.DL/I
  - → Ad hoc query for DL/I databases
- Automatic code generators:
  1. Cross-System Product (CSP) supports:
     a) CICS/ESA with IMS/DB
     b) IMS/TM to access DL/I DBs and/or DB2 tables
  2. Application Development Facility II (IMSADF II)

Figure 5-33. IMS Related Products (CM011565)

Notes:

Summary

High transaction volumes, fast response.

Very large terminal networks.

Recovery/Restart procedures for transaction network.

Figure 5-34. Summary (CM011567)

Notes:

Unit 6. IMS Transaction Message Processing

What This Unit Is About

Application program structure to use the IMS Transaction Manager.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss the elements of a call to retrieve or send information to an IMS terminal

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 2 activities

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8017-00 IMS/ESA V5 Application Programming: TM

Objective

- Discuss the elements of a call to retrieve or send information to an IMS terminal.

Figure 6-1. Objective (CM011601)

Notes:

6.1 IMS Transaction Message Processing

IMS/TM Message Flow

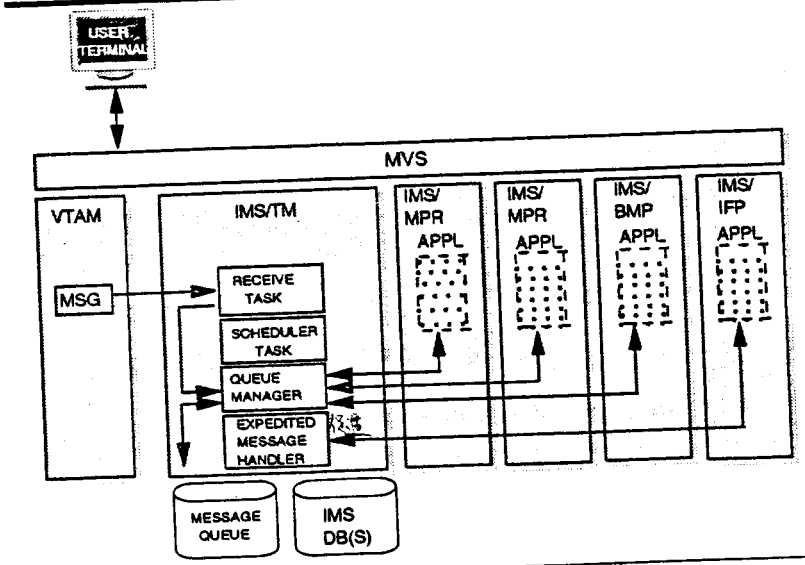


Figure 6-2. IMS/TM Message Flow (CM011600)

Notes:

IMS/TM Message Flow ...

From application-programmer's point-of-view:

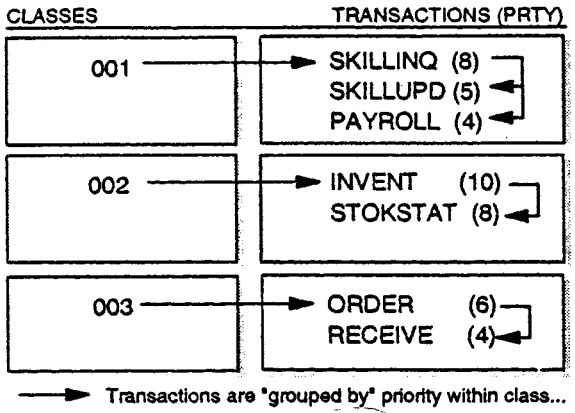
- Phased Application Implementation
  - → Add new databases, programs, transactions without disrupting production system
- Standard Application Programming Interface (API)
  - Common DL/I Interface used for calls:
    - → Database and message queue
  - Allows program independence of terminal types
  - Independent of programming-language
    - → COBOL, PL/I, ASM, FORTRAN, PASCAL, C, etc.
- Capable of supporting large-volume transaction workload
  - → How many transactions/second with "good" response?

Figure 6-3. IMS/TM Message Flow... (CM011601)

Notes:

Message Classes

- TRANSACTIONS are assigned two (2) attributes:
  1. Class, and
  2. Priority:



- Class and priority can be changed via /ASSIGN command

Figure 6-4. Message Classes (CM011603)

Notes:

Message "Scheduling"

- **Application programs are automatically scheduled only into Message Processing Regions (MPRs):**

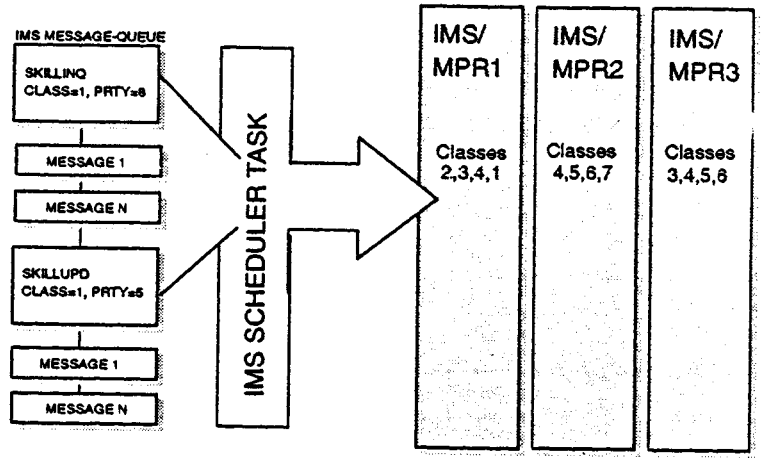


Figure 6-5. Message "Scheduling" (CM011605)

Notes:

Transaction Scheduled

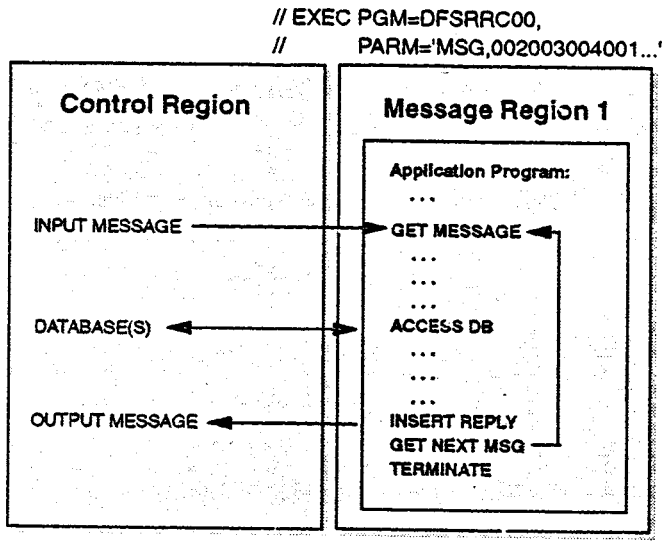


Figure 6-6. Transaction Scheduled (CM011607)

Notes:

COBOL Coding Example

```
READ THE INCOMING MESSAGE:

CALL 'CBLTDLI' USING GU, IOPCB, IO-AREA.


CHECK THE STATUS CODE:

IF STATUS-CODE = 'QC'
     END THE PROGRAM - OUT OF MESSAGES

HANDLE THE REQUEST:
BLAH. BLAH. BLAH.

SEND THE REPLY:

CALL 'CBLTDLI' USING ISRT, IOPCB, IO-AREA.

LOOP TO THE TOP TO GET THE NEXT MESSAGE.
```

Figure 6-7. COBOL Coding Example (CM011609)

Notes:

Message Queue "Limit Priority"

- **Additional factors affecting scheduling...**

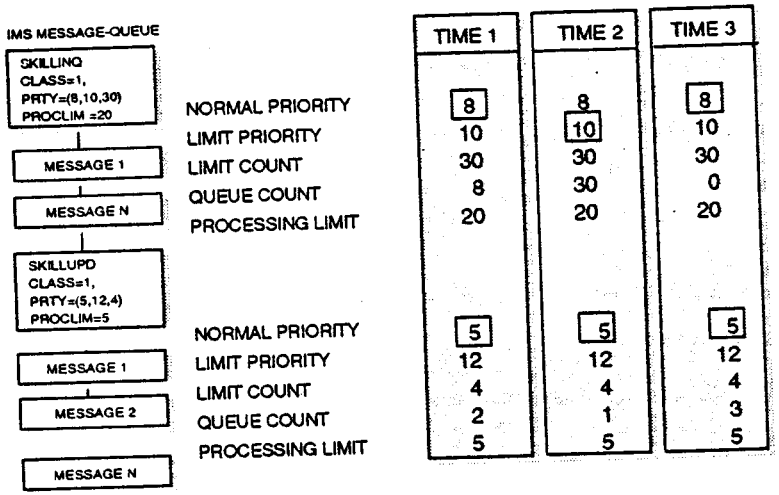


Figure 6-8. Message Queue "Limit Priority" (CM011611)

Notes:

Message Region Options

More scheduling options...

- "Wait-For-Input (WFI)" transactions:
  - → Allows a program to remain scheduled to wait for the next transaction, subject to PROCLIM=
- Parallel-scheduling:
  - → Allows the same program to be scheduled in parallel address-spaces at the same time PARLIM
    1. If processing a different transaction code
    2. Subject to queue-count of each transaction code
    3. Upper limit maximum
- Preload of application programs into certain regions
  - → Program is "ready and waiting" when application schedules

Figure 6-9. Message Region Options (CM011613)

Notes:

Dynamic Transaction Backout

When the program reads the incoming message:

Call 'CBLTDLI' using GU, IOPCB...

All updates and messages for the previous transaction are committed.

Dynamic transaction backout is invoked automatically when a program abends.

Any uncommitted updates or messages are purged from the system.

Figure 6-10. Dynamic Transaction Backout (CM011615)

Notes:

Defining Message Processing

IMSGEN macros:

```
APPLCTN   PSB=INQPGM,                    <-- Class
          PGMTYPE=(TP,,1)
TRANSACT  CODE=SKILLINQ,
          PRTY=(8,10,30),
          PROCLIM=20

APPLCTN   PSB=UPDPGM,                    <-- Class
          PGMTYPE=(TP,,1)
TRANSACT  CODE=SKILLUPD,
          PRTY=(5,12,4),
          PROCLIM=5

TRANSACT  CODE=SKILLDLT,
          PRTY=(8,8,20)

APPLCTN   PSB=ORDPGM,                    <-- Class
          PGMTYPE=(TP,,4)
TRANSACT  CODE=ORDERENT,
          PRTY=(8,12,4),
          SPA=(200,CORE)

APPLCTN   PSB=BILLPGM,                   <-- Class
          PGMTYPE=(TP,,7)
TRANSACT  CODE=BILLING,
          PRTY=(12,14,5),
          SPA=(200,CORE)
```

Figure 6-11. Defining Message Processing (CM011617)

Notes:

Message Processing Programs (MPPs)

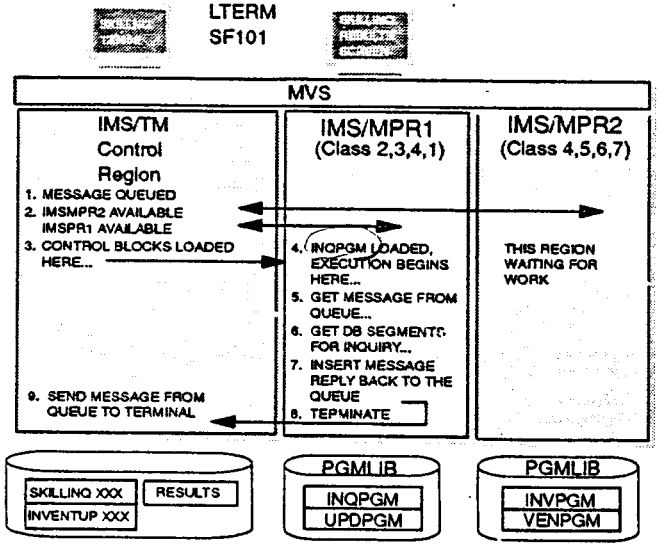


Figure 6-12. Message Processing Programs (MPPs) (CM011619)

Notes:

Message Processing Calls

FUNCTION	PCB	CODE
RETRIEVING MESSAGES:		
GET UNIQUE	IO-PCB	GUbb
GET NEXT	IO-PCB	GNbb
SENDING MESSAGES to ORIGINATING TERMINAL:		
INSERT	IO-PCB	ISRT
SENDING MESSAGES to ALTERNATE TERMINALS:		
CHANGE	ALT-PCB	CHNG
INSERT	ALT-PCB	ISRT
PURGE	ALT-PCB	PURG
CHECKPOINTING of the BATCH APPLICATION:		
CHECKPOINT	IO-PCB	CHKP
SYNC-POINT	IO-PCB	SYNC

Figure 6-13. Message Processing Calls (CM011621)

Notes:

Conversational Processing

- IMS maintains a "Scratch-Pad Area (SPA)" between schedulings of "conversational" programs

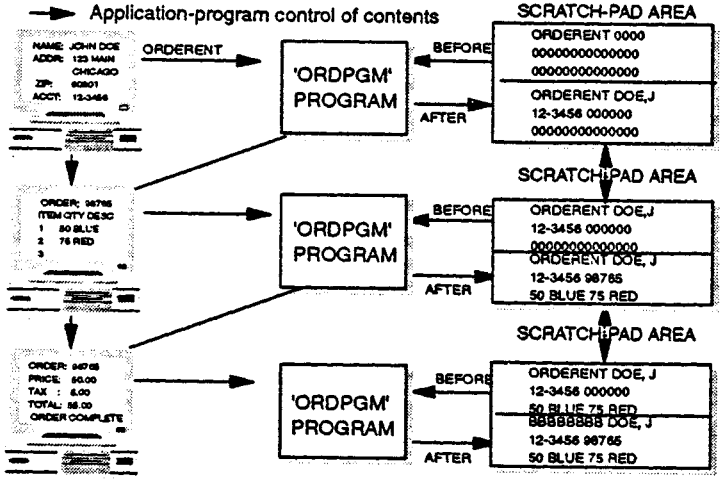


Figure 6-14. Conversational Processing (CM011623)

Notes:

AIB Interface Application Interface Block

Most DL/I calls can be issued in 2 ways:

Call 'AIBTDLI' using GU, AIB, IO-AREA...

or

Call 'CBLTDLI' using GU, PCB, IO-AREA...

PSBGEN may optionally assign a name to a PCB.
Application can reference by name rather than address.

Figure 6-15. AIB Interface Application Interface Block (CM011625)

Notes:

Batch Message Processing (BMP)

- Programs are "batch" programs

  Submitted when ready by Operations

- May or may not access the message queue:

1. Message-driven BMP: processes input messages and inserts output reply

   BMP address-space JCL parameters (IN= , OUT=)

2. Non-message-driven BMP: access databases only

IMSGEN macros:

```
APPLCTN   PSB=HQPGM,
          PGMTYPE=(BATCH)

TRANSACT  CODE=HQUPDT,
          PRTY=(0)

APPLCTN   PSB=DB2PGM,
          PGMTYPE=(BATCH)
```

Figure 6-16. Batch Message Processing (BMP) (CM011633)

Notes:

Batch Message Processing (BMP) ...

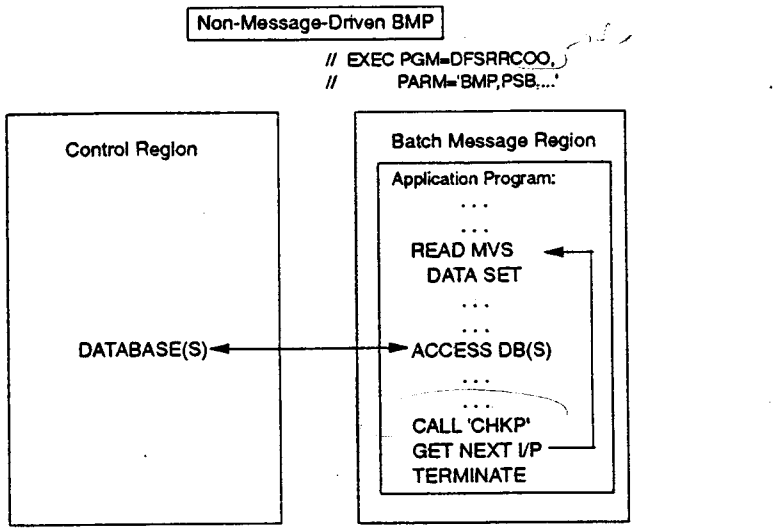


Figure 6-17. Batch Message Processing (BMP)... (CM011635)

Notes:

Application Program Recovery/Restart

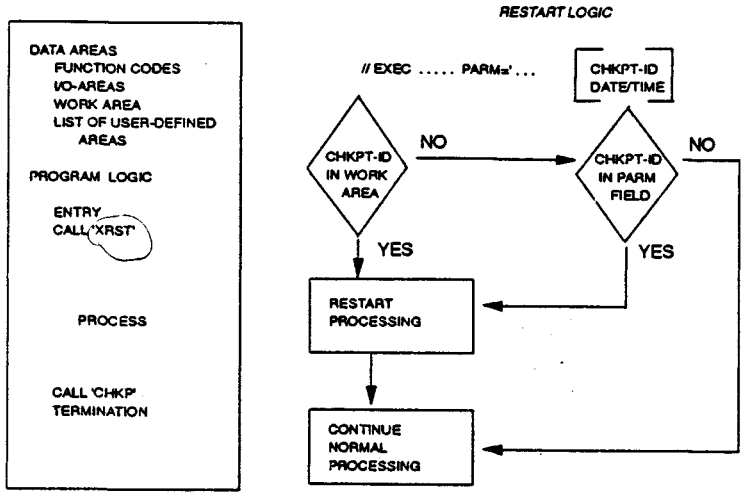


Figure 6-18. Application Program Recovery/Restart (CM011637)

Notes:

CHKP Checkpoint & XRST Extended Restart

CHKP
CHECKPOINT

Call 'CBLTDLI' using CHKP, IOPCB,...

Commit updates
Free program isolation locks
Save working storage (optional)

XRST
EXTENDED RESTART

Call 'CBLTDLI' using XRST, IOPCB,...

Repositions Databases - including GSAM
Recover working storage (optional)

Figure 6-19. CHKP Checkpoint (CM011639)

Notes:

IMS/ESA APPC Explicit API Flow

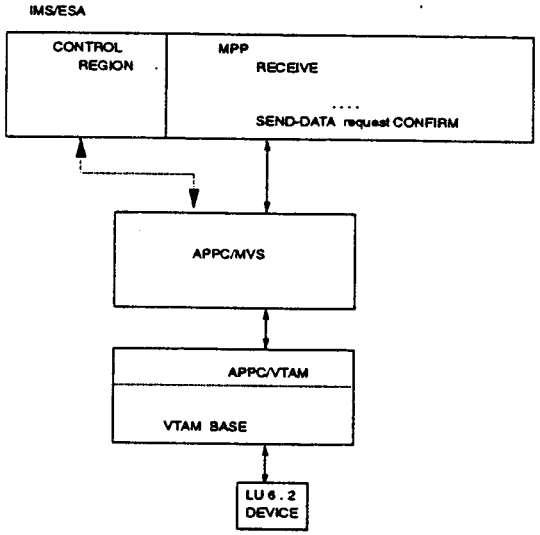


Figure 6-20. IMS/ESA APPC Explicit API FLow (CM011641)

Notes:

APPC Support

- APSB (allocate PSB) call
  - Dynamically allocate a PSB
  - CPIC driven programs have no access to a PSB when scheduled
  - Allows access to IMS resources when application is CPIC driven
    - DL/I data bases
    - Alternate PCB output
- All transaction modes supported for current & modified programs
  - Conversational
  - Response
  - Non-response

Figure 6-21. APPC Support (CM011643)

Notes:

Summary

- Application programming similar to IMS/DB
- Transaction and database logging not a programming responsibility
- APPC programming for distributed processing

Figure 6-22. Summary (CM011642)

Notes:

Unit 7. Interactive Fast Path

What This Unit Is About

The IMS Fast Path Database access methods and the Expedited Message Handler.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe a Data Entry Database (DEDB) and reasons for its use.
- Describe a Main Storage Database (MSDB) and reasons for its use.
- List the processing differences between Fast Path and Message Queue processing.

How You Will Check Your Progress

Accountability:

- Lab exercise summarizing Day 2 activities

References

GC26-3467-00 IMS/ESA V5 General Information IMS/ESA V5 General Information

SC26-8014-00 IMS/ESA V5 Administration Guide: TM

SC26-8017-00 IMS/ESA V5 Application Programming: TM

Interactive Fast Path

What is Fast Path?

1. **Data entry databases (DEDBs):**
   - HDAM-like
   - Areas and partitioning
   - Multiple Area Data Sets (MADS)
   - VSO option
2. **Main Storage Databases (MSDBs):**
   - Table-like
   - Resident in main storage
     - ➡ no call-related I/Os...
3. **Expedited Message Handler (EMH):**
   - Simplified message flow
     - ➡ Bypasses normal IMS/TM message queue
   - Simplified scheduling
   - Program is "waiting for input..."

Figure 7-1. Interactive Fast Path (CM011700)

Notes:

7.1 Data Entry Database (DEDB)

Interactive Fast Path

What is Fast Path?

1. **Data entry databases (DEDBs):**
   - HDAM-like
   - Areas and partitioning
   - Multiple Area Data Sets (MADS)
   - VSO option
2. **Main Storage Databases (MSDBs):**
   - Table-like
   - Resident in main storage
     - ➡ no call-related I/Os...
3. **Expedited Message Handler (EMH):**
   - Simplified message flow
     - ➡ Bypasses normal IMS/TM message queue
   - Simplified scheduling
   - Program is "waiting for input..."

Figure 7-2. Interactive Fast Path. (CM011700)

Notes:

Record Structure

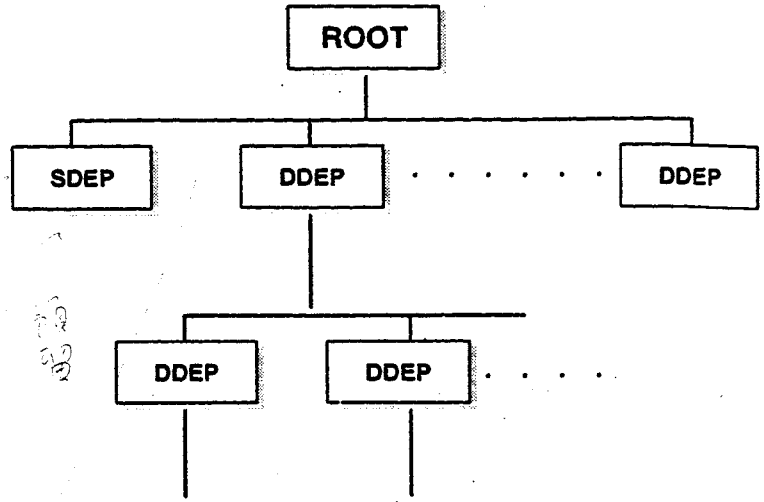


Figure 7-3. Record Structure (CM011703)

Notes:

- Similar to HDAM
  - Uses Randomizer
  - 15 Levels and 127 Segment Typess
- Sequential Dependent (SDEP)
  - Only 1 SDEP type in structure; multiple occurrences allowed)
  - No HDAM equivalent
  - Very fast ISRT and DLET
  - Fast Data Capture
- Direct Dependent (DDEP)
  - Like HDAM dependents
  - Subset Pointers
- Not Supported
  - Secondary Indexes
  - Logical Relationships

DEDB Storage Format

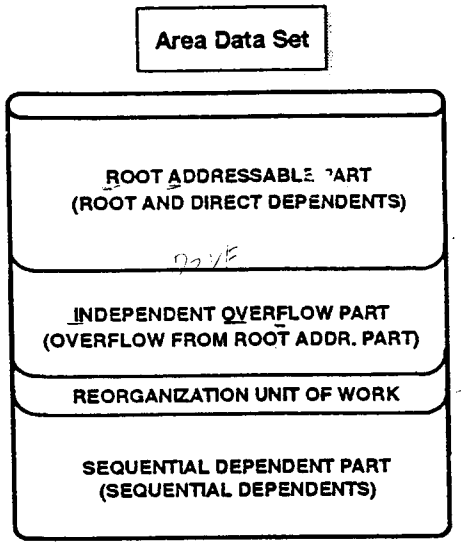
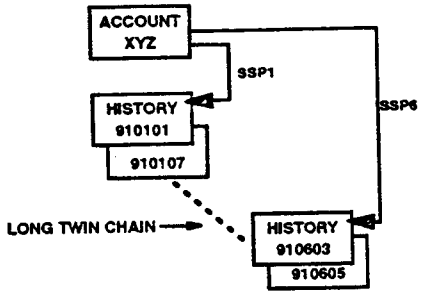


Figure 7-4. DEDB Storage Format (CM011705)

Notes:

Subset Pointer

- Keep six months of history segments:



- For Direct Dependents
  - May define 0 to 8 Subset Pointers
- User program Control of Pointers
  - Use - Set - Move - Zero
- Application:
  - To find first June history segment, search bypasses January through May!

Figure 7-5. Subset Pointer (CM011707)

Notes:

SDEP Efficiency

- **Efficiency comes from:**
  1. ISRT: Multiple inserts per I/O
  2. SCAN: Retrieve only SDEPs
  3. DELETE: no I/Os
- **Ideal for "Temporary" data**
  - Audit Trail
  - Holding Area
  - Journalizing
  - Memo/Post

Figure 7-6. SDEP Efficiency (CM011709)

Notes:

Database Partitioning

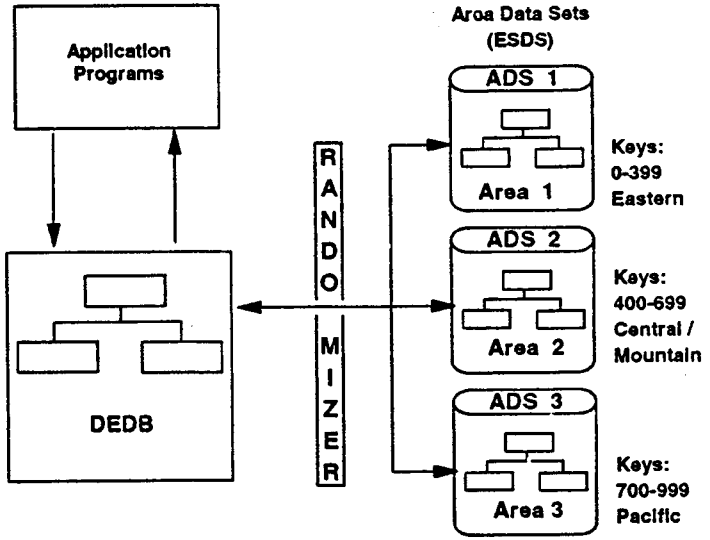


Figure 7-7. Database Partitioning ... (CM011711)

Notes:

Database Partitioning

- DEDBs can be divided into multiple partitions called areas
- Each area is an Area Data Set (ADS)
  - 4 Gigabytes per Area
- Up to 240 Areas per DEDB
  - 960 GB Database (1 trillion bytes)
    - ➡ 4K for Every Person in U.S.
- Single Database Image
  - Areas Transparent to Application
  - Area determined by Randomizer
    - Based on root key

Figure 7-8. Database Partitioning (CM011713)

Notes:

Database Partitioning

Why Partition a Database?

- Too large for Single Data set
  - Exceeds 4 GB Limit
  - Want to keep data set size small
- To improve performance
  - Spread database activity across multiple data sets
  - Parallel applications do not contend
- To improve availability
  - Areas are independently-managed
    - Allocated, opened, closed, recovered, reorganized
  - Application continues even if all Areas are not available

Figure 7-9. Database Partitioning... (CM011715)

Notes:

Multiple Area Data Sets (MADS)

- **An Area can be replicated up to 7 copies**
- **Dynamically Created**
  - ➡ Area available to on-line update during create
- **Provides protection against**
  - I/O errors
  - Device/path unavailability

Figure 7-10. Multiple Area Data Sets (MADS) (CM011717)

Notes:

Multiple Area Data Sets (MADS)

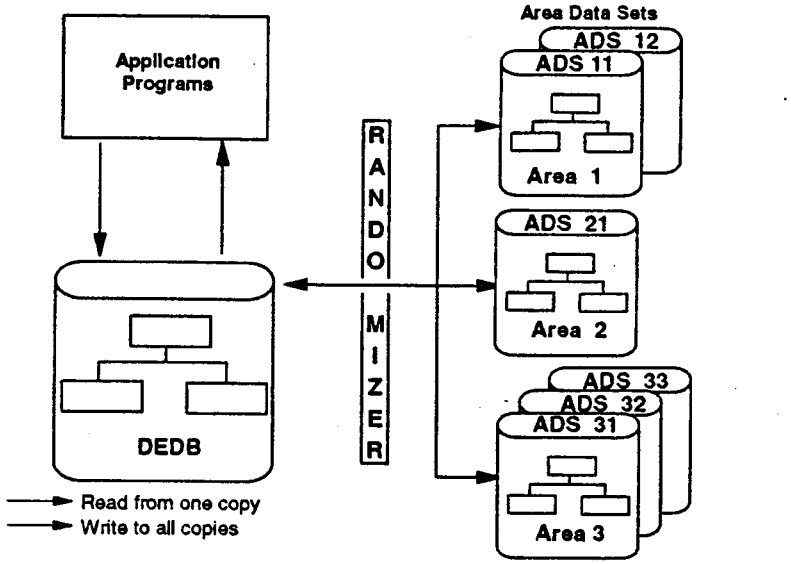


Figure 7-11. Multiple Area Data Sets (MADS) (CM011719)

Notes:

Online DEDB Utilities

- Direct reorganization with concurrent update activity
- Scan/delete for Sequential Dependents (SDEPs)
- Area Data Set (ADS) create:
  - ➡ Creates new ADS(s) from one or more existing ADS(s)
  - ➡ Can be used for Online Recovery from I/O Errors
- Concurrent Image Copy (CIC) Option:
  - Modified to Use MADS Input
  - Supported by DBRC

Figure 7-12. Online DEDB Utilities (CM011721)

Notes:

Online DEDB Utilities

- Online Recovery

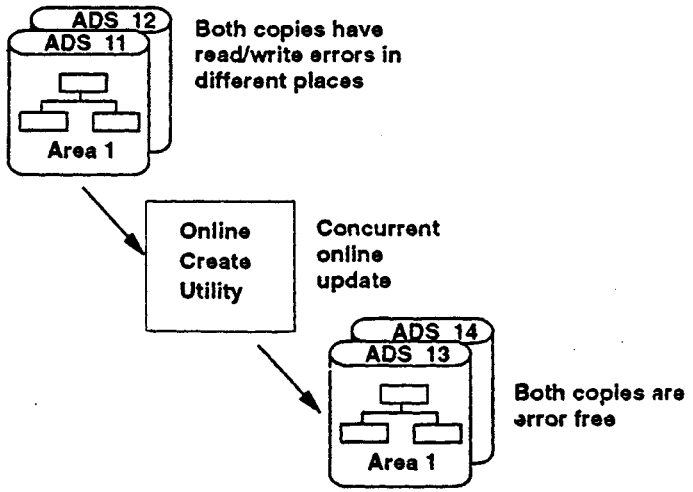


Figure 7-13. Online DEDB Utilities (CM011723)

Notes:

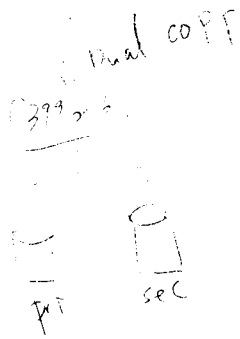
High-Speed Sequential Processing

(HSSP)

- Designed to improve sequential processing of DEDBs
- Private buffer pools
- Optimized locking
- Reduced logging
- Supported wherever DEDBs are supported

Figure 7-14. High-Speed Sequential Processing (CM011725)

Notes:



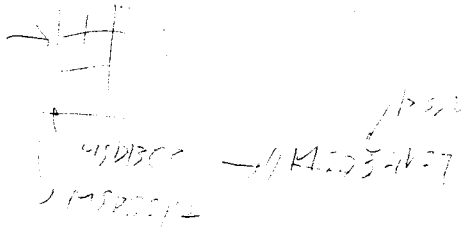
Fast Path Virtual Storage Option

VSO

- VSO objective is to eliminate the need for MSDBs
- One or more DEDBs may be mapped into virtual storage
- Less read I/O
- Less lock contention
- Fewer physical writes
- Only previous solution was MSDB

Figure 7-15. Fast Path Virtual Storage Option (CM011727)

Notes:



Field Call

- Single call both searches and updates the record
- Uses an FSA:

  Field Search Argument
- "If account 123 has a balance of at least $100.00 deduct $100.00 and update the record"
- Similar to SSA
- Previously only for MSDBs

Figure 7-16. Field Call (CM011729)

Notes:

DEDB Summary

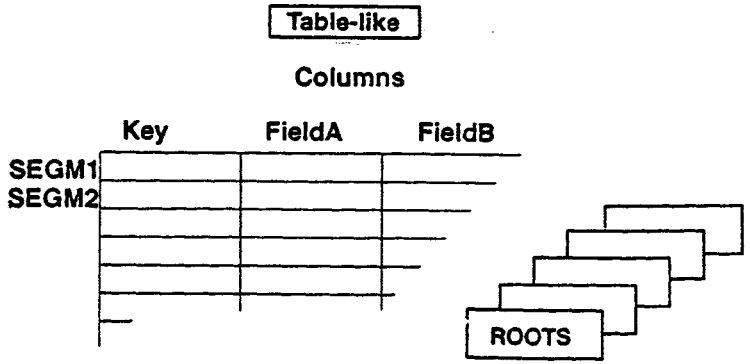
- Subset Pointers for less I/O
- Sequential dependents
  - More efficient insert
  - More efficient mass retrieval
  - More efficient mass delete
- Area concept
  - More parallelism possible
  - Multiple Area Data Sets (MADS)
- Benefits
  - High performances
  - High Availability
  - Continuous operations
  - Large data volumes

Figure 7-17. DEDB Summary (CM011731)

Notes:

7.2 Main Storage Database (MSDB)

Main Storage Database (MSDB)



- Root-only database
- Fixed length segments (32000 max)
- Direct or Sequential Access
  - ➡ GU or GN calls

Figure 7-18. Main Storage Database (MSDB) (CM011733)

Notes:

MSDB Types

- **Terminal Related:**
  - LTERM name is key
  - Updated by input terminal only
  - Insert and delete allowed
- **Non-terminal Related:**
  - User-defined key
  - Can be updated by any user
  - Insert and Delete not allowed

Figure 7-19. MSDB Types (CM011735)

Notes:

MSDB Processing

- **Loaded at IMS Initialization**
  - ➡ Long-Term page-fixed
- **No call-related I/O: MSDBs are always in storage**
- **Full integrity/recovery**
  - "Image copy" at each System Checkpoint to distinct MSDB checkpoint data sets (MSDBCPn)
- **Standard DL/I call interface**
  - GU, GN, REPL, ...
  - FLD (Field Call)

Figure 7-20. MSDB Processing (CM011737)

Notes:

Other MSDB Examples

Table Look-up

- Examples:
  - Rates/Fares
  - Schedules
  - Itineraries
  - Prices
  - Tax Tables

Statistics

- Examples:
  - Number of Transactions by terminal
  - Number of Transactions by Time of Day
  - Number of Transactions by Div/Dept/Office/...
  - Number of Orders by Customer/Geo. Area/...

Figure 7-21. Other MSDB Examples (CM011739)

Notes:

MSDB Summary

- No I/O Activity
  - ➡ The "fast" in Fast Path...
- Shorter Path Length
  - Standard Calls
  - Field Calls
- Reduced Contention
  - No DASD Contention
  - Reduced Enqueue
- VSO Option similar in DEDB

Figure 7-22. MSDB Summary (CM011741)

Notes:

7.3 Expedited Message Handler (EMH)

Expedited Message Handler (EMH)

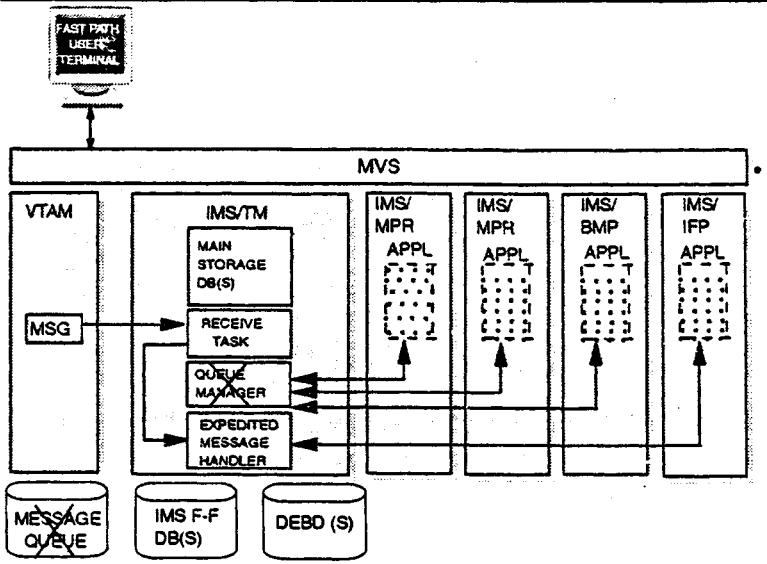


Figure 7-23. Expedited Message Handler (EMH) (CM011743)

Notes:

Fast Path Application Program

- Defined during System Definition

```
APPLCTN PSB=IFPAPPL,...,FPATH=YES
```

- Runs in "IFP" Address-space
  - Waits-For-Input
  - Parallel scheduling allowed
  - Same program can run in MSG region with a different name
- Processes only Fast Path transactions

Figure 7-24. Fast Path Application Program (CM011745)

Notes:

Fast Path Transactions

1. Fast Path *Exclusive* Transactions:
   - Process only in IFP region

```
APPLCTN PSB=IFPAPPL,...,FPATH=YES
TRANSACT CODE=FPTRAN1,...
RTCODE CODE=FPRTCD1
```

2. Fast Path *Potential* Transactions:
   - Process in IFP or MSG region
   - *Exit Routine* makes determination

```
APPLCTN PSB=MSGAPPL,...
TRANSACT CODE=FFTRAN2,...,FPATH=YES
```

Figure 7-25. Fast Path Transactions (CM011747)

Notes:

Fast Path Terminals

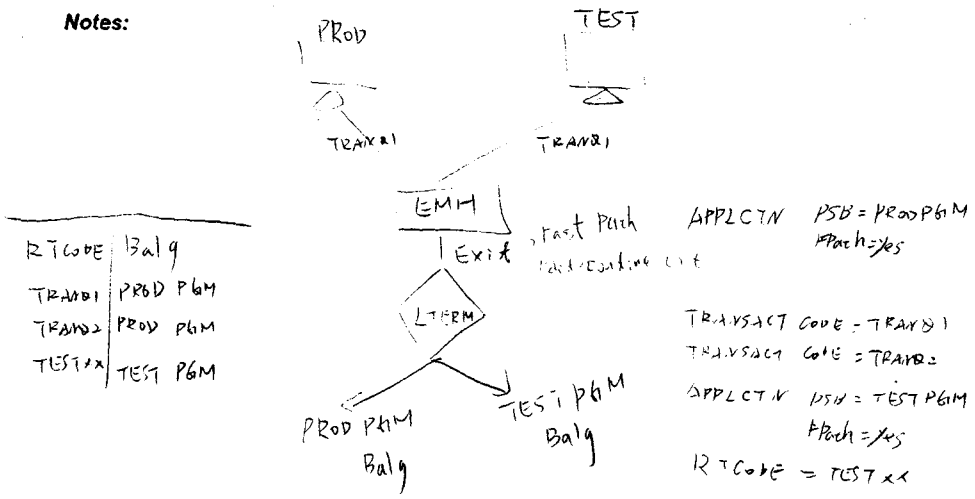
- Fast Path transactions can only be entered from FP-eligible terminals

  TERMINAL ....,FPBUF=nnnn

- Dedicated Buffer
- Single segment
- Response mode
- Non-conversational
- Scheduling is FIFO by PSB

Figure 7-26. Fast Path Terminals (CM011749)

Notes:



Message Flow

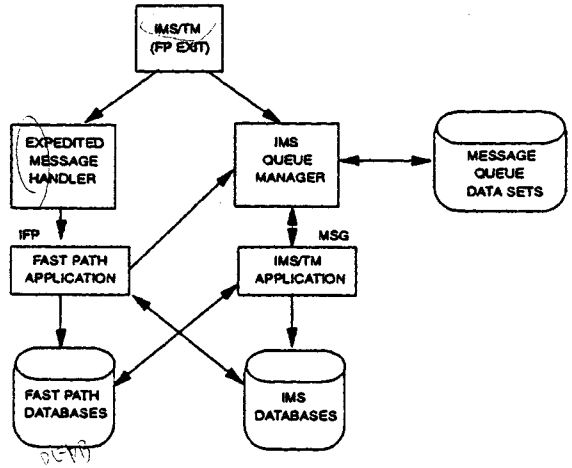


Figure 7-27. Message Flow (CM011751)

Notes:

EMH Summary

- Simplied Message Handling
  - Dedicated buffers
  - Nomessage queue data sets
- Simplied Scheduling
  - No priorities (FIFO)
  - Wait-for-input
  - Multiple transaction-codes per program schedule
- Less Logging Overall
  - Fewer log record types
  - After images only

Figure 7-28. EMH Summary (CM011753)

Notes:

More Options

- Databases
  - HDAM, HIDAM, HISAM
  - DEDB, MSDB
- Message Queuing and Scheduling
  - IMS Message Queue Manager
  - Expedited Message Handler

Figure 7-29. More Options (CM011755)

Notes:

Unit 8. IMS Large-System Environment

What This Unit Is About

How IMS operates in a large scale environment where data must be available to many systems concurrently.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss different methods of sharing data between systems.
- Describe the use of "backup" IMS systems to ensure non-stop operation of critical IMS networks.

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8014-00 IMS/ESA V5 Administration Guide: TM

Large System Environment

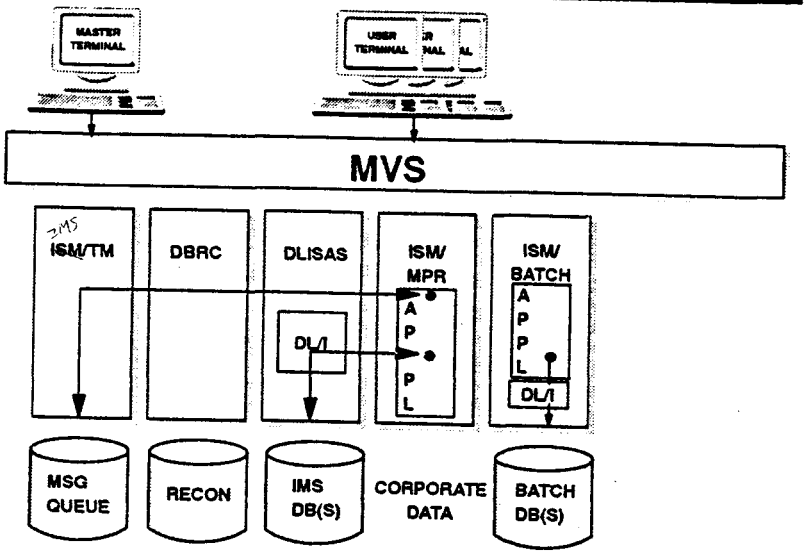
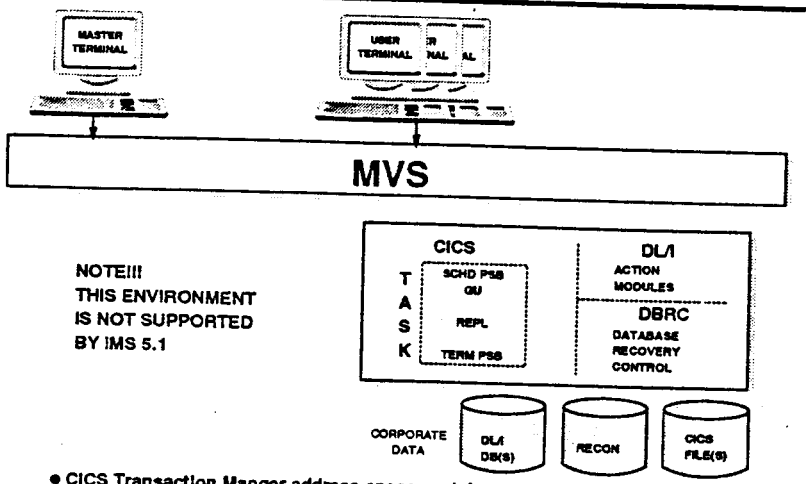


Figure 8-1. Large - System Environment (CM011800)

Notes:

8.1 IMS DBCTL with CICS

CICS Local DL/I Environment



NOTE!!!
THIS ENVIRONMENT
IS NOT SUPPORTED
BY IMS 5.1

- CICS Transaction Manger address-space contains:
  1. DL/I modules
  2. DBRC modules

Figure 8-2. CICS Local DL/I Environment (CM011801)

Notes:

CICS IRC/ISC Environments

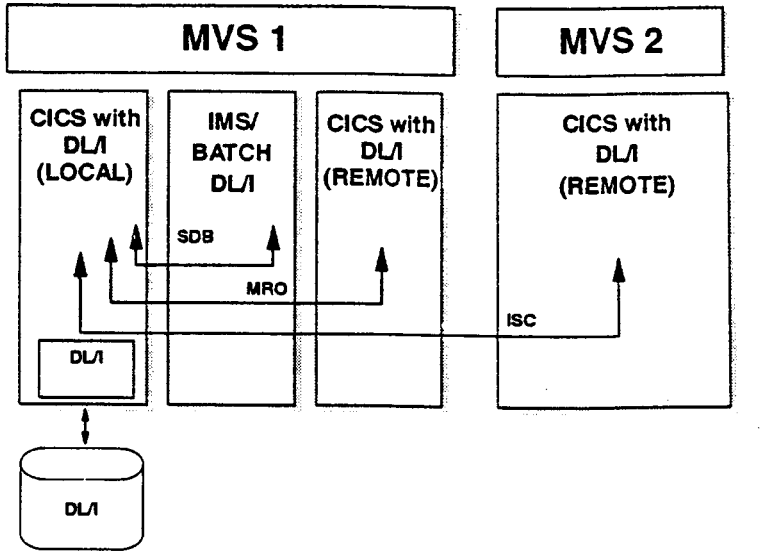


Figure 8-3. CICS IRC/ISC Environments (CM011803)

Notes:

SDB = Shared Database
MRO = Multiple Region Option
IRC = Inter-Region Communication
ISC = Inter-System Communication

Database Control (DBCTL) Subsystem

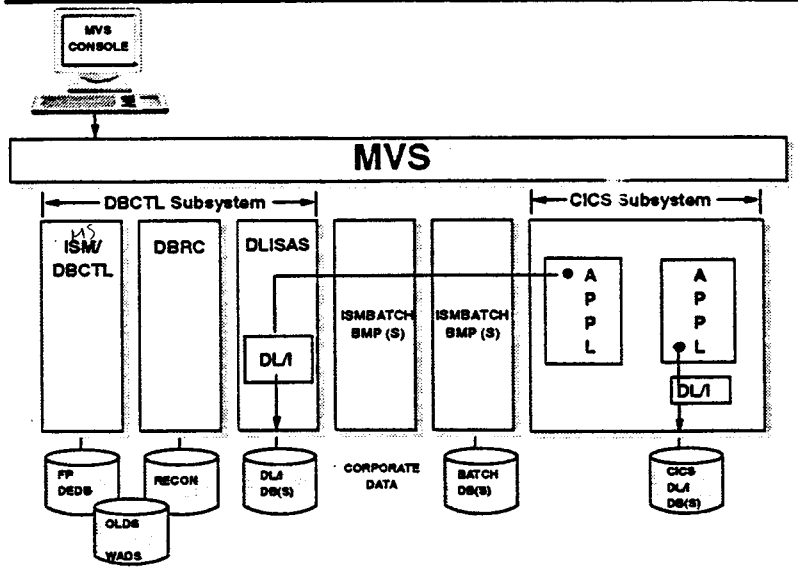
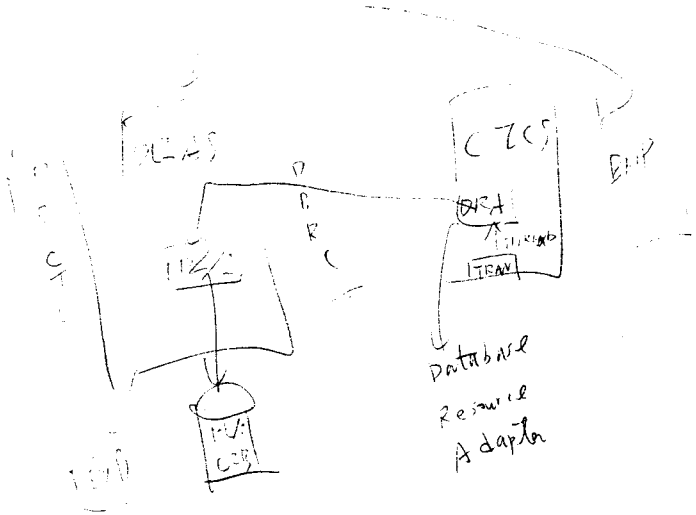


Figure 8-4. Database Control (DBCTL) Subsystem (CM011805)

Notes:



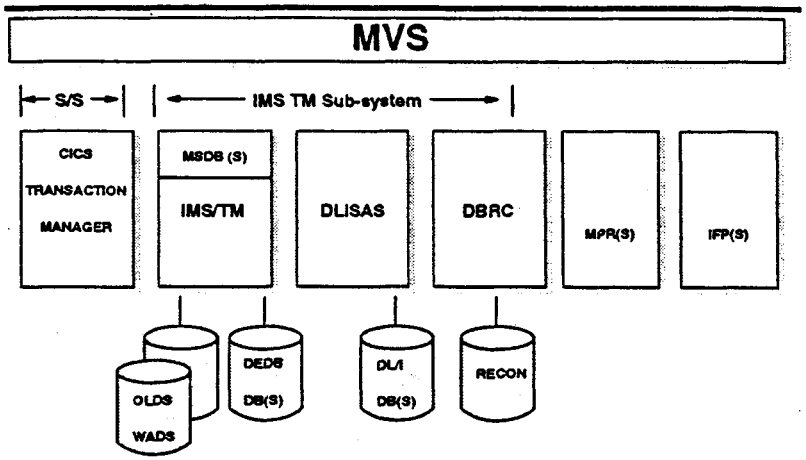
Database Control (DBCTL) Subsystem

- An IMS/DB stand-alone subsystem
- Supports Full-Function databases and Fast Path DEDBs
- Maximum 1,000 concurrent BMP Address-spaces
  - Total BMPs + CICS transaction "threads"
- Convert off-line batch jobs to BMPs
- BMP address-spaces may contain:
  - Batch programs (converted DBBBATCH/DLIBATCH)
  - Online Image Copy (OLIC) jobs
  - Fast Path DEDB utilities
- MVS Console for operator control:
  - START, SHUTDOWN, RESTART of DBCTL
  - Connection from CICS to DBCTL
- Concurrent connections from multiple CICS sub-systems

Figure 8-5. Database Control (DBCTL) Subsystem... (CM011807)

Notes:

IMS TM used for DBCTL Services



- IMS TM subsystem provides DBCTL Services
  - ➡ Connected CICS(s) have concurrent database access without Data Sharing (IRLM not needed)
  - ➡ MSDBs are only known to IMS TM transactions

Figure 8-6. IMS TM used for DBCTL Services (CM011809)

Notes:

DBCTL Environment

- To fully utilize the DBCTL environment some application program changes may be required - especially in a CICS only installation.
- "DLI" type batch jobs must be modified to work in the "BMP" environment.
- A typical program might require changes as follows:
  - PSBGEN - COMPAT=Y - GETS "DUMMY" IOPCB
  - PROGRAM CODE
    - PROGRAM ENTRY - IOPCB
    - CALL "CBLTDLI" USING CHKP...

Figure 8-7. DBCTL Environment (CM011811)

Notes:

CICS-DBCTL Summary

- Performance
  - Up to 1,000 concurrent applications
- Separate subsystems and address-spaces provide VSCR
- Common logging facility over CICS journal:
  - Dual logging option
  - User-specified number of log data sets
  - Automatic archiving under DBRC control
  - Combined logging versus separate off-line batch job logging

Figure 8-8. CICS-DBCTL Summary (CM011813)

Notes:

CICS-DBCTL Summary

- Increased data availability:
  - Concurrent Image Copy (CIC) of DB Datasets
  - DB data available after DBCTL restart
    - WRITE I/O virtual buffers are logged
  - Backout failure: Stop only those DBs not backed-out
  - Successful PSB schedule even when some DBs are Not Available
- Fast Path DEDB access:
  - High availability (Area concept)
  - Potential performance benefits
  - Larger databases

Figure 8-9. CICS-DBCTL Summary... (CM011815)

Notes:

8.2 IMS Data Sharing

Data Sharing Definition

"The DATA SHARING facility allows multiple application programs in different IMS and /or CICS (/A PGM) subsystems to concurrently process common DL/I databases."

1. DATABASE-LEVEL SHARING
   - DBRC controls used of database
2. BLOCK-LEVEL DATA SHARING
   - IMS Resource Lock Manager (IRLM) provides "global" Program Isolation (P-I)

Figure 8-10. Data Sharing Definition (CM011817)

Notes:

Data Sharing Definition

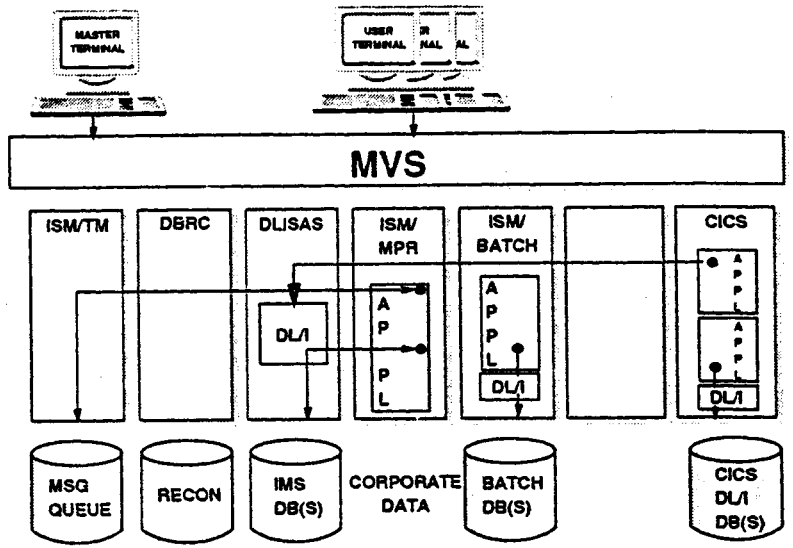


Figure 8-11. Data Sharing Definition... (CM011819)

Notes:

DBRC Share-Level Options

- DATABASE-LEVEL SHARING:
  1. SHARELVL(0)
     - Guarantees EXCLUSIVE use
  2. SHARELVL(1)
     - 1 updater; multiple Read-Only (RO)
     - Multiple readers, RO and/or Read w/Integrity (RD)
- BLOCK-LEVEL DATA SHARING:
  1. SHARELVL(2)
     - Multiple UP/RD/RO
     - Maximum 1 MVS system image
     - INTRA-system, DBRC, one IRLM
- SHARELVL(3)
     - Multiple UP/RD/RO
     - MULTIPLE MVS system images
     - INTER-system, two IRLMs, VTAM "connection"

Figure 8-12. DBRC Share-Level Options (CM011821)

Notes:

Database-Level Sharing

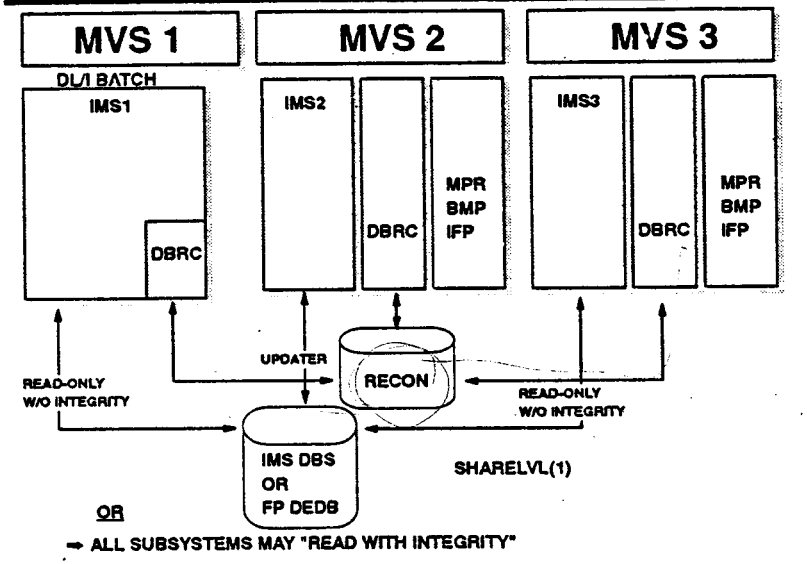


Figure 8-13. Database-Level Sharing (CM011823)

Notes:

Intra-System Block-Level Data Sharing

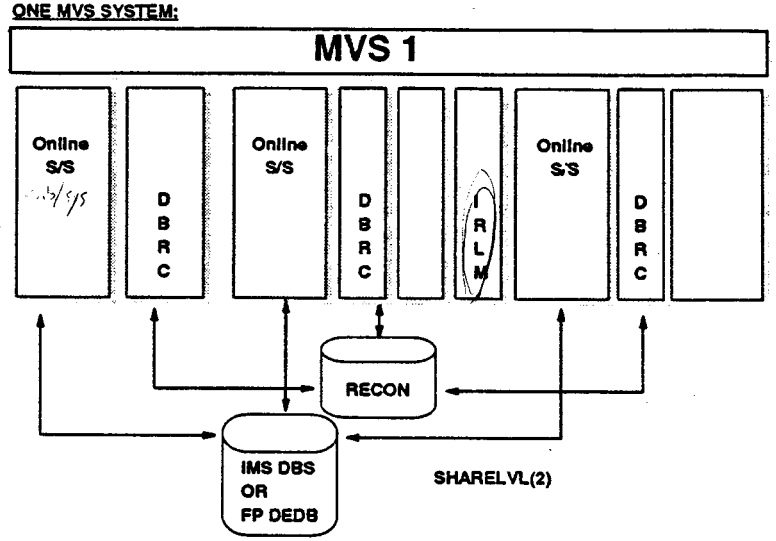


Figure 8-14. Intra-System Block-Level Data Sharing (CM011825)

Notes:

Inter-System Block-Level Data Sharing

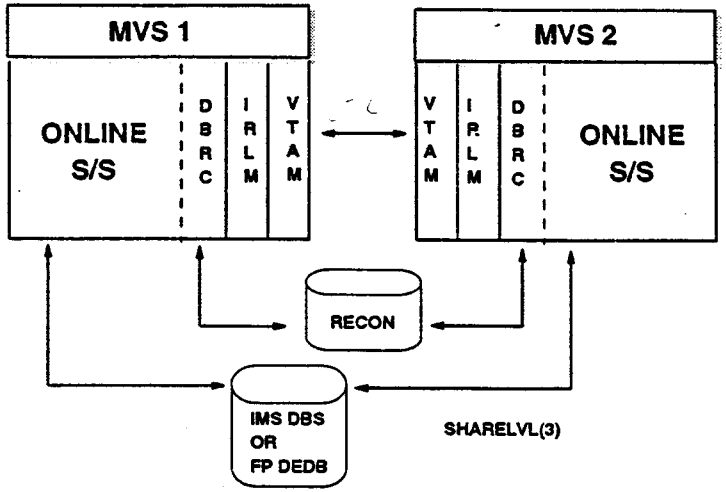


Figure 8-15. Inter-System Block-Level Data Sharing (CM011827)

Notes:

IMS In A Parallel Sysplex

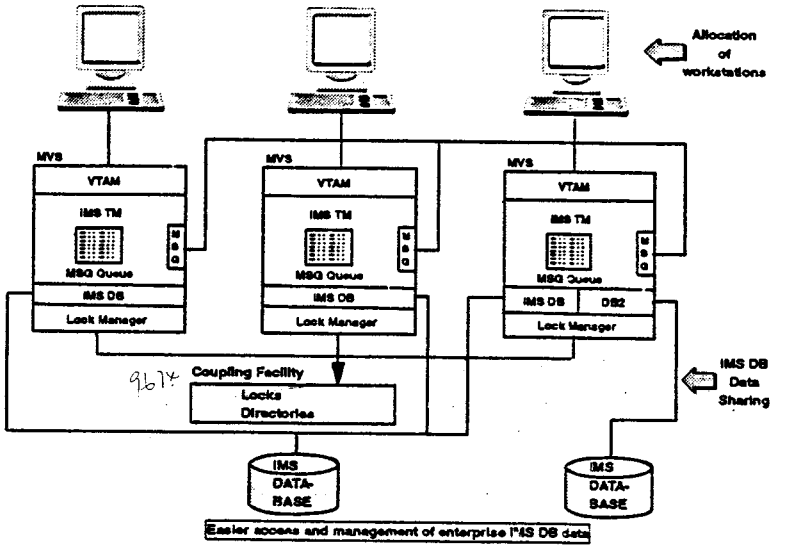


Figure 8-16. IMS in a Parallel Sysplex (CM011833)

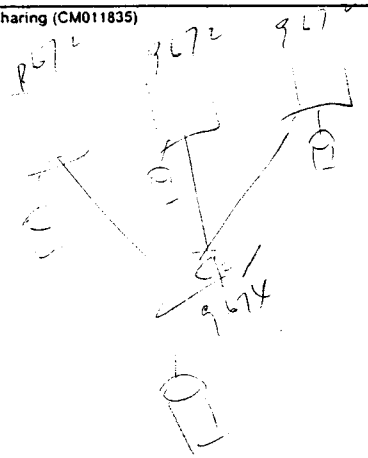
Notes:

N-Way Data Sharing

- IMS/ESA Version 5 allows up to 32 IMS subsystems distributed among, at most, 32 MVS images, in an MVS sysplex, to share databases
  - Sharing across more than 2 MVS images requires the use of the S/390 Parallel Transaction Server (PTS) Coupling Facility technology
- IMS/ESA Version 5 contains a new IRLM (Version 2.1)
  - IRLM Version 2.1 can communicate with up to 31 other instances of IRLM Version 2.1, using the S/390 PTS Coupling Facility technology
  - Old IRLM (Version 1.5) also included with IMS/ESA Version 5
- Data sharing is supported among all IMS/ESA versions
  - Block level data sharing logic (what locks are obtained when) has not changed in IMS/ESA Version 5

Figure 8-17. N-Way Data Sharing (CM011835)

Notes:



Data Sharing Summary

- Other Data Sharing subsystems could be:
  - IMS DBCTL
  - IMS TM
  - Off-Line Batch (DBBBATCH/DLIBATCH)
  - CICS w/ "Local DL/I"
  - Selected IMS utilities, such as CIC
- IRLM is required for Block-Level Data Sharing
- Increased system availability:
  - More operational flexibility via separate subsystems
  - Failure isolation via separate subsystems

Figure 8-18. Data Sharing Summary (CM011831)

Notes:

8.3 IMS Extended Recovery Facility (XRF) and Remote Site Recovery (RSR)

Typical IMS Restart XRF (Extend Recovery Facility)

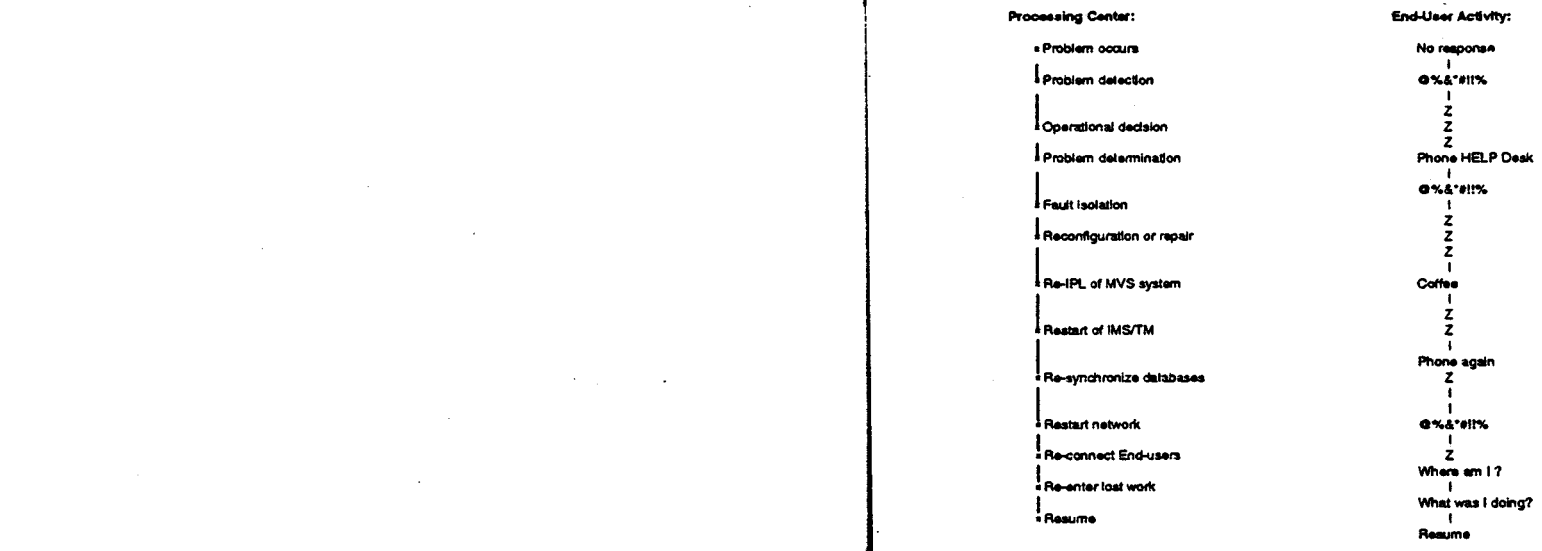


Figure 8-19. Normal Recovery/Restart (CM011837)

Notes:

Steady State: Before Takeover

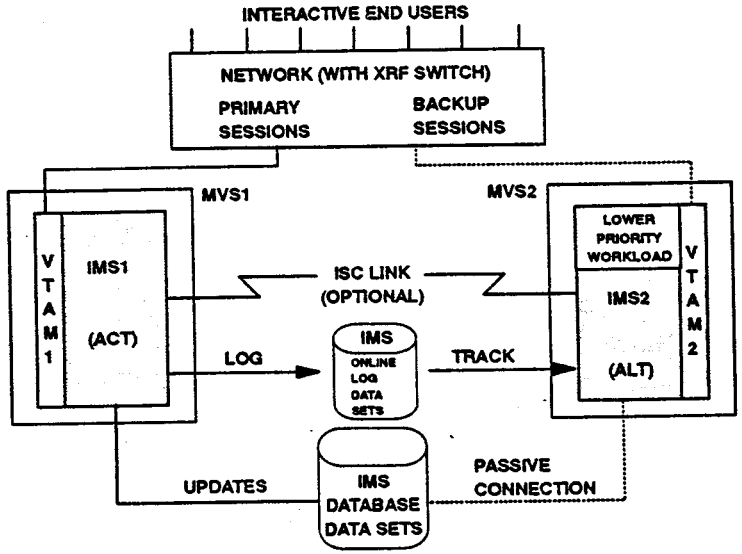


Figure 8-20. Steady State: Before Takeover (CM011839)

Notes:

During Takeover

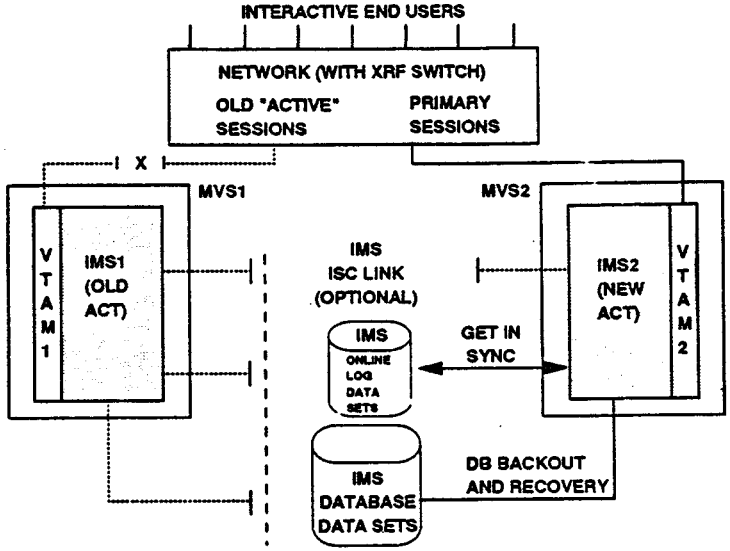


Figure 8-21. During Takeover (CM011841)

Notes:

After Takeover

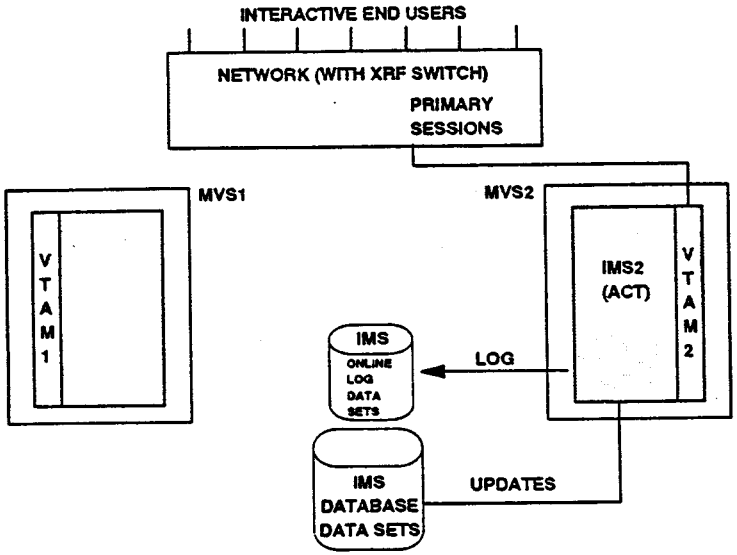


Figure 8-22. After Takeover (CM011843)

Notes:

Steady State: After Takeover

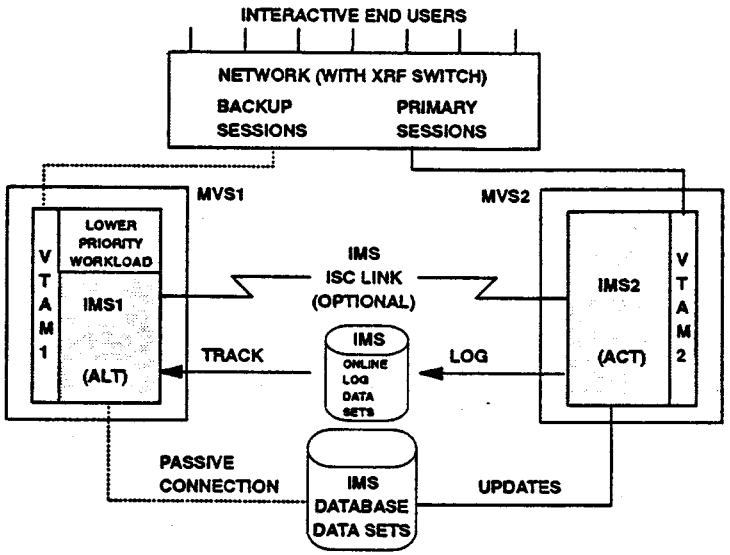


Figure 8-23. Steady State: After Takeover (CM011845)

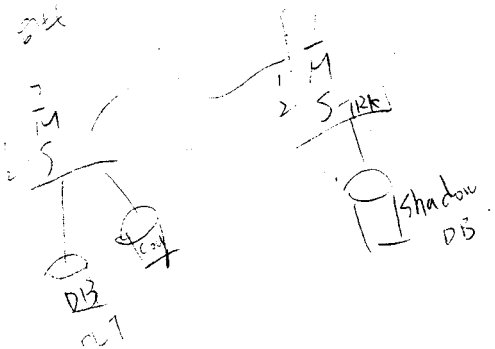
Notes:

Remote Site Recovery

- RSR Database Support
  - Database Level Tracking (DLT)
    - RSR provides remote database shadowing
    - Reduces service outage to minutes
    - Requires dedicated DASD for databases while tracking
- Recovery Level Tracking (RLT)
  - User must recover databases following a takeover
  - Service outage will be measured in hours
  - Does not require dedicated DASD for databases while tracking
- Level of support selectable by database
- Level of support defined in DBRC RECONs and can be changed dynamically

Figure 8-24. Remote Site Recovery (CM011847)

Notes:



Summary

Extended Recovery Facility (XRF) for non-stop system

Remote Site Recovery (RSR) for backup systems at remote locations.

Figure 8-25. Summary (CM011849)

Notes:

Unit 9. IMS Distributed Processing

What This Unit Is About

Sharing of a transaction load between multiple IMS systems or IMS and CICS system

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe IMS Multiple Systems Coupling (MSC).
- Describe IMS/CICS/VTAM Inter System Communication (ISC).
- Discuss the differences between MSC and ISC.

References

GS26-3467-00 IMS/ESA V5 General Information

SC26-8014-00 IMS/ESA V5 Administration Guide: TM

Distributed Processing

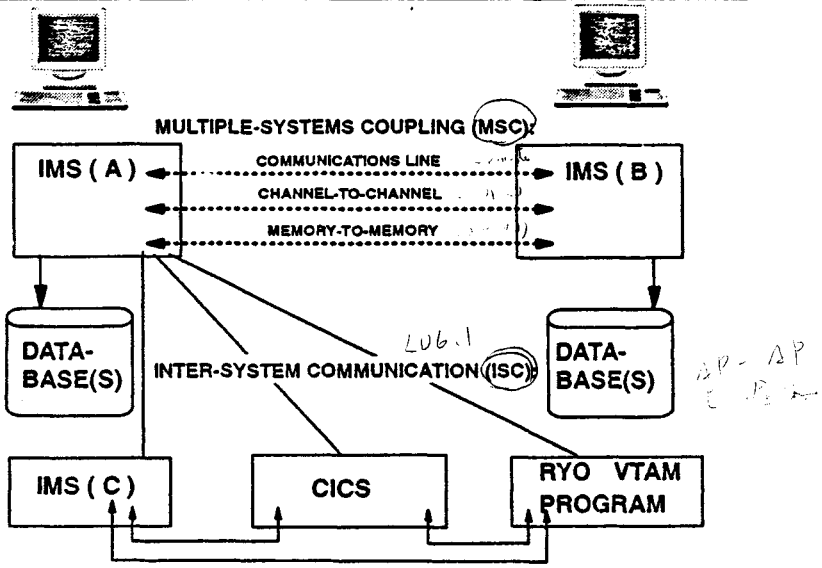


Figure 9-1. Distributed Processing (CM011900)

Notes:

9.1 Multiple-Systems Coupling (MSC)

Multiple-Systems Coupling (MSC)

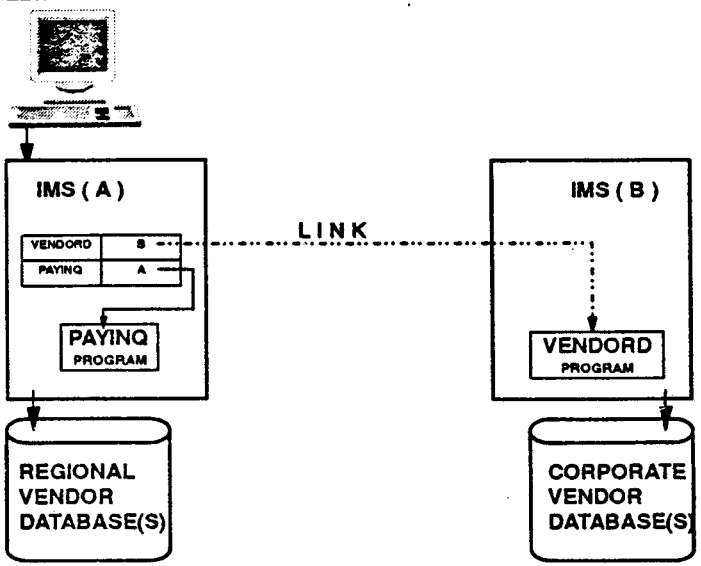


Figure 9-2. Multiple-Systems Coupling (MSC) (CM011903)

Notes:

Multiple-Systems Coupling (MSC)...

- **Allows for the distribution of IMS transactions between IMS subsystems**
  - ➡ Transactions entered from one IMS subsystem may be processed on another IMS subsystem
- **IMS-to-IMS configuration only**
  - ➡ Each system is a complete IMS subsystem
- **Transactions may be "routed" via:**
  1. Binary synchronous (BSC) line with BTAM
  2. SNA VTAM SDLC Link
  3. Channel-to-Channel (CTC) Adapter
  4. Main Storage-to-Main Storage (MTM) Link (within one MVS image)
- **Routing is automatic via IMSGEN process**
  - ➡ **No application-program or end-user need be involved**

Figure 9-3. Multiple-Systems Coupling (MSC)... (CM011905)

Notes:

Distributed Databases

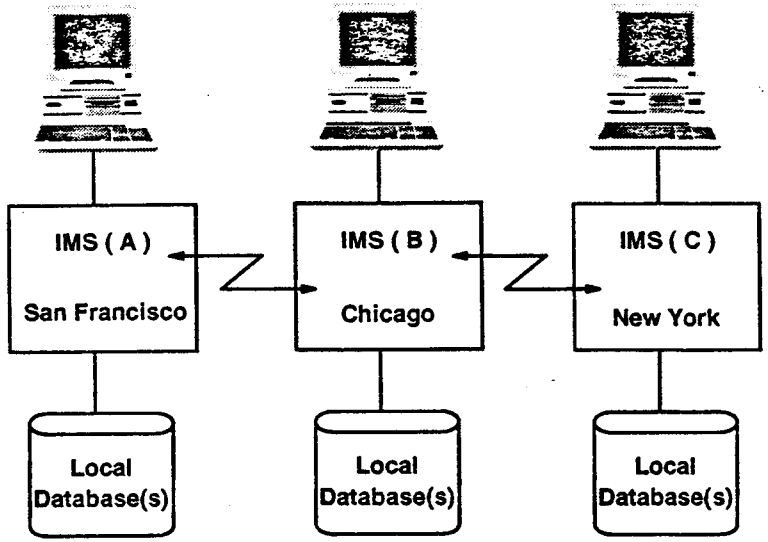


Figure 9-4. Distributed Databases (CM011907)

Notes:

Distributed Applications

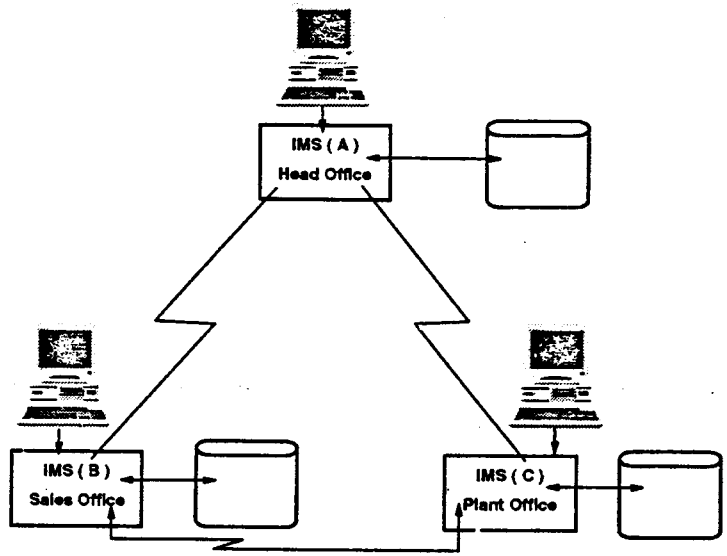


Figure 9-5. Distributed Applications (CM011909)

Notes:

Increased Capacity

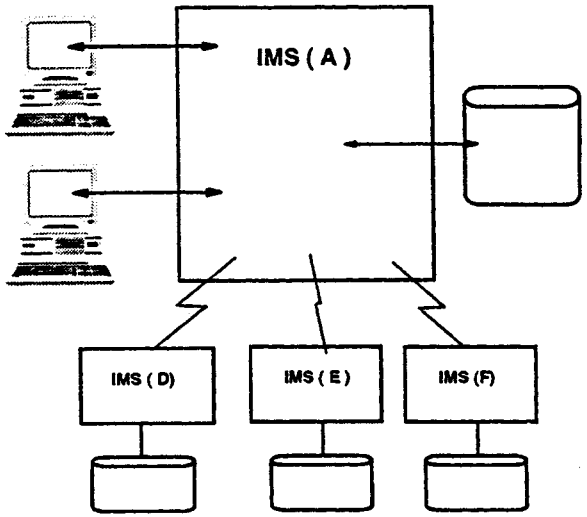
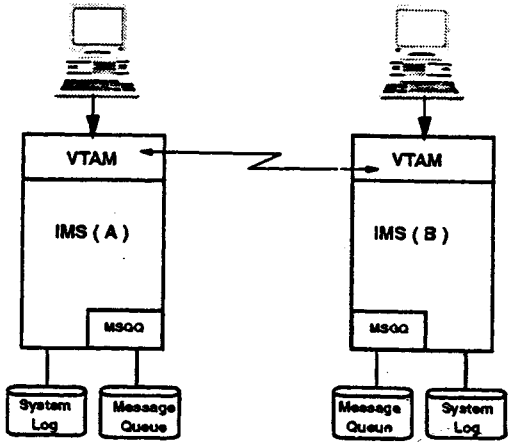


Figure 9-6. Increased Capacity (CM011913)

Notes:

MSC Message Routing



- Destinations are modifiable via user exits:
  - Too many, or duplicate names
  - Destinations chosen dynamically
- Each subsystem has its own MTO and OLDSs

WLR

Figure 9-7. MSC Message Routing (CM011917)

Notes:

9.2 Inter-System Communication (ISC)

Inter-System Communication (ISC)

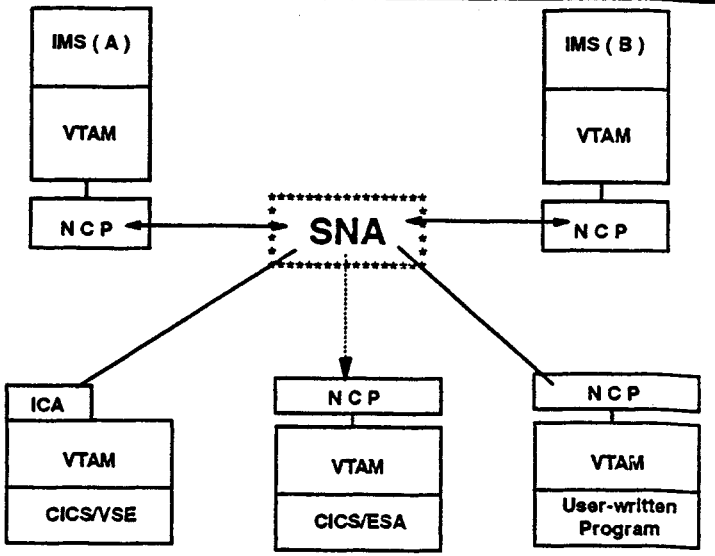


Figure 9-8. Inter-System Communication (ISC) (CM011919)

Notes:

Inter-System Communication (ISC)...

- ISC is part of MSC
  - But MSC is IMS-to-IMS only
- ISC allows more flexible configuration than MSC
  - IMS-to-IMS
  - IMS-to-CICS
  - IMS-to-"RYO" VTAM program

Figure 9-9. Inter-System Communication (ISC)... (CM011921)

Notes:

ISC Message Routing

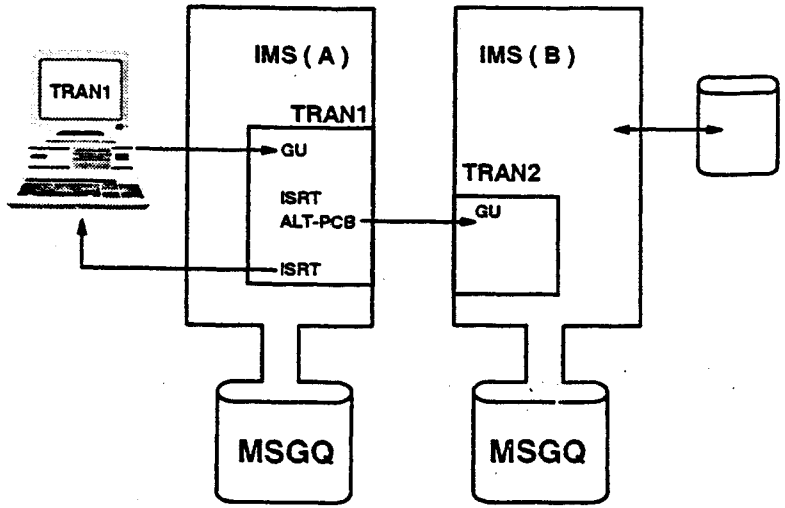


Figure 9-10. ISC Message Routing (CM011925)

Notes:

ISC Message Routing...

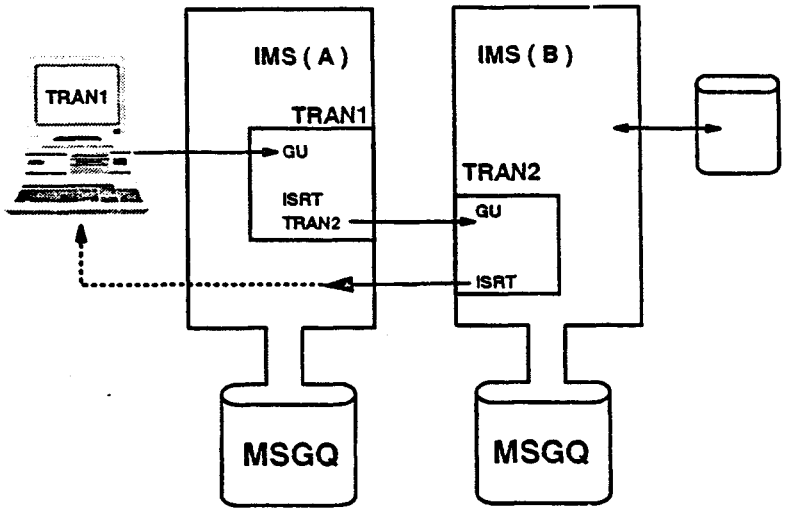


Figure 9-11. ISC Message Routing (CM011927)

Notes:

CICS-IMS Connectivity

- CICS-to-IMS scenarios:

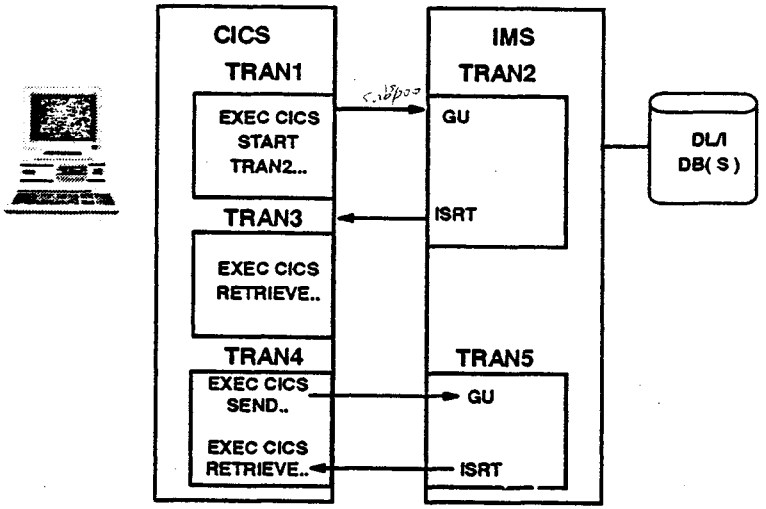


Figure 9-12. CICS-IMS Connectivity (CM011931)

Notes:

CICS-IMS Connectivity...

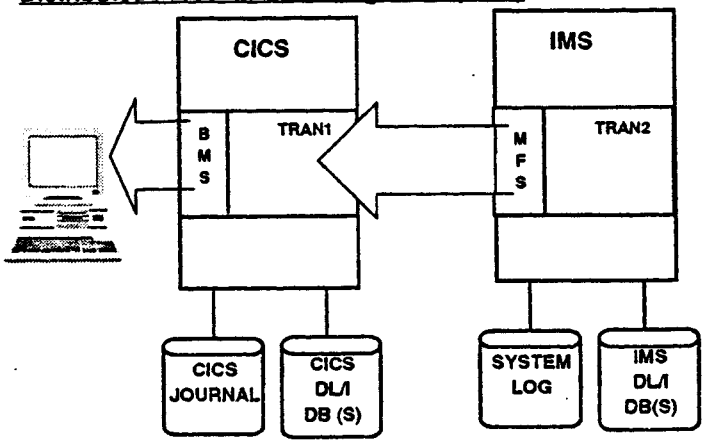


Figure 9-13. CICS-IMS Connectivity (CM011933)

Notes:

DCCTL Region Type

Like an IMS Transaction Manager Region without the databases

Two primary uses:

- Use as a router in a multiple system configuration such as MSC
- A system in which all databases activity is Non-IMS/DB

Figure 9-14. DCCTL Region Type (CM011939)

Notes:

Summary

- Describe IMS Multiple Systems Coupling (MSC).
- Describe IMS/CICS/VTAM Inter System Communication (ISC).
- Discuss the differences between MSC and ISC.

Figure 9-15. MSC versus ISC Summary... (CM011941)

Notes:

Summary

- Describe IMS Multiple Systems Coupling (MSC).
- Describe IMS/CICS/VTAM Inter System Communication (ISC).
- Discuss the differences between MSC and ISC.

Figure 9-16. MSC Vs ISC Summary (CM011941)

Notes:

MSC Vs ISC Summary...

MSC Functions:	ISC Functions:
MSC connects multiple IMS subsystems only to each other. These IMS systems may all reside in one processor, or they may reside in different processors.	ISC can connect either like or unlike subsystems, as long as the connected subsystems both implement ISC. A user may couple an IMS subsystem to: 1) Another IMS subsystem 2) A CICS subsystem 3) A user-written subsystem
Communication in the MSC environment is from subsystem to subsystem. MSC is really switching a message from one IMS message queue to another IMS message queue.	Communication is between application programs in the two subsystems. The subsystems themselves are session partners, supporting logical flow between the applications.
Processing is transparent to the user. That is, MSC processing appears as if it is occurring in a single system.	Because ISC permits coupling of unlike subsystems, message routing requires involvement by the terminal user or the application to determine the message destination. Specified routing parameters may be overridden, modified, or deleted by MFS.
When not using the MSC directed routing capability, the terminal operator or application does not need to know routing information. Routing is automatic, based on the system definition parameters.	ISC provides a unique message switching capability that permits message routing to occur without involving a user application; or else the user application implements switching logic.
MSC permits the steps of a conversation to be distributed over multiple IMS subsystems, transparent to both the source terminal operator and to each conversational step in the application.	ISC does not support conversational processing between two IMS subsystems; but, IMS conversational processing is supported from a CICS or RYO subsystem.
MSC does not support Fast Path. The Expedited Message Handler (EMH) does not have access to the message queue, which is needed by MSC	ISC permits the use of Fast Path between IMS subsystems.
MFS support is separated in sharing subsystems.	ISC permits using MFS in an IMS subsystem to assist the routing and formatting of messages between subsystems.

Figure 9-17. MSC Vs ISC Summary (CM011937)

Notes:

Unit 10. IMS-DB2 Attachment

What This Unit Is About

Access to DB2 data in the IMS Transaction Manager environment.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss the differences between relational (DB2) and hierarchical (IMS) access methods.
- Describe the steps required to access DB2 data from an IMS/TM system.
- Describe use of the Data Propagator.

References

GC26-3467-00 IMS/ESA V5 General Information

SC26-8014-00 IMS/ESA V5 Administration Guide: TM

Objectives

- Discuss the difference between relational (DB2) and hierarchical (IMS) access methods.
- Describe the steps required to access DB2 data from an IMS/TM system,
- Describe use of the Data Propagator.

Figure 10-1. Objectives (CM011a01)

Notes:

10.1 IMS-DB2 Attachment

Multiple Database Managers

- Data is accessible from multiple environments:

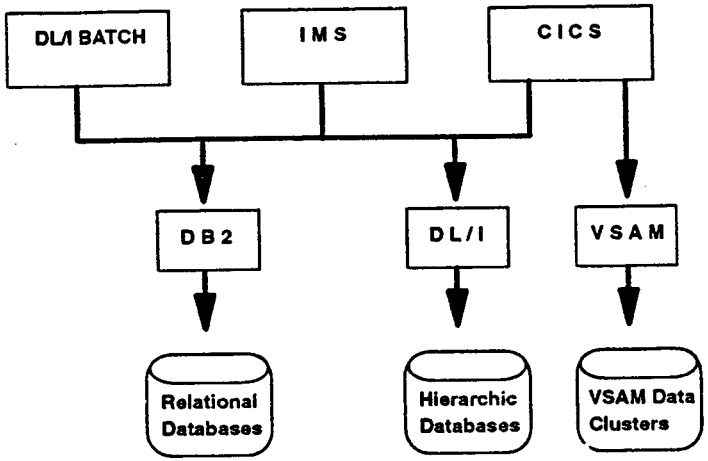


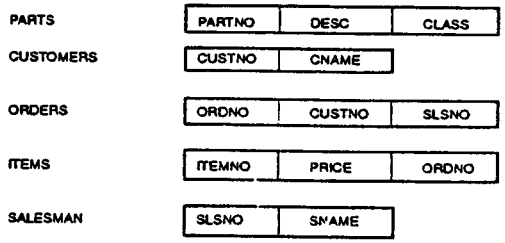
Figure 10-2. Multiple Database Managers (CM011A00)

Notes:

What Is DB2 ?

DB2 Relational Model

- Relations are conceptually tables of "flat" files



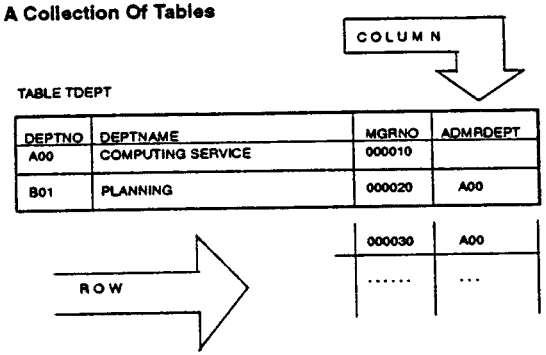
- All information is represented by field values
- Records are related by field values, independent of storage structure:
  - A Record is a Row of a Table
  - A Field is a Column in a Row

Figure 10-3. What is DB2 ? (CM011A01)

Notes:

What Is A Relational Database ?

- A "Relational Database" Is Perceived Externally As A Collection Of Tables



- All Data Relationships Are Represented By Field Values
- DB2 Tables Are Defined And Manipulated Using The SQL Language

Figure 10-4. What is A Relational Database ? (CM011A03)

Notes:

DB2 Interactive (DB2I)

- DB2I Is A DB2 TSO/ISPF Interface For Application Development And Testing:

DB2I PRIMARY OPTION MENU

Select one of the following DB2 functions and press ENTER.

1	SPUFI	(Process SQL statements)
2	DCLGEN	(Generate SQL and source language declarations)
3	PROGRAM PREPARATION	(Prepare a DB2 application program to run)
4	PRECOMPILE	(Invoke DB2 precompiler)
5	BIND/REBIND/FREE	(BIND, REBIND, or FREE for application plans)
6	RUN	(Run an SQL program)
7	DB2	(Issue DB2 commands)
8	UTILITIES	(Invoke DB2 utilities)
9	DB2I DEFAULTS	(Set global parameters)
X	EXIT	(Leave DB2I)

PRESS: END TO EXIT HELP FOR MORE INFORMATION

Figure 10-5. DB2 Interactive (DB2I) (CM011A05)

Notes:

DB2 Interactive (DB2I) ...

PROGRAMMING AND TESTING USING DB2I:

- HIGH PRODUCTIVITY THROUGH DB2I
  - INTERACTIVE TESTING OF SQL
  - EASY CREATION OF TEST TABLES AND DATA
  - ONLINE HELP AND TUTORIAL
- DECLARATIONS GENERATION
  - COBOL AND PL/I DECLARATIONS FOR TABLES, VIEWS, AND DATA STRUCTURES
- PROGRAM PREPARATION
  - PRE-COMPILE
  - COMPILE AND LINK-EDIT
  - BIND
  - EXECUTION

Figure 10-6. DB2 Interactive (DB2I)... (CM011A07)

Notes:

Using Query Management Facility

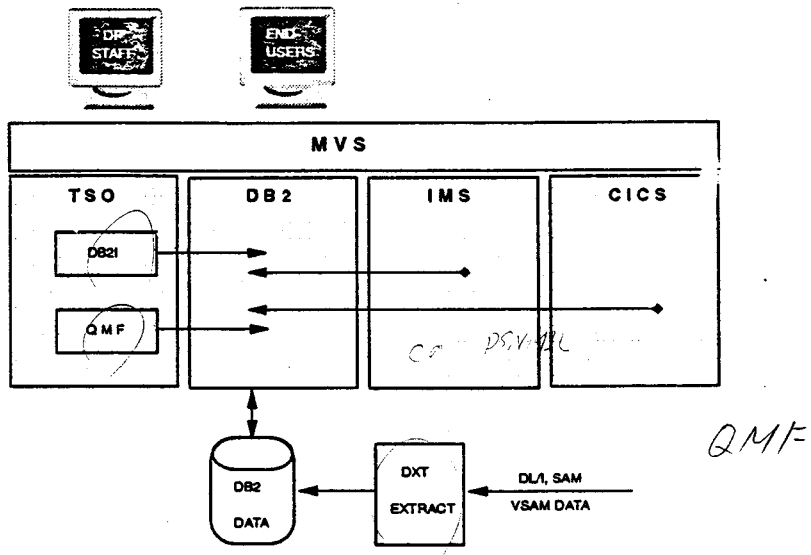


Figure 10-7. Using Query Management Facility (QMF) (CM011A09)

Notes:

IMS or DB2?

HIERARCHIC	RELATIONAL
IMS	**DB2**
PHYSICAL/LOGICAL IMPLEMENTATION ALLOWS VERY EFFICIENT PROCESSING FOR SOME QUESTIONS.	ANY QUESTION SUPPORTED BY DATA CAN BE ASKED - I/O MAY BE A CONCERN.
SUPPORTS VERY HIGH TRANSACTION RATES	POWERFUL QUERY LANGUAGE

Figure 10-8. IMS or DB2 ? (CM011A11)

Notes:

10.2 Defining the Environment

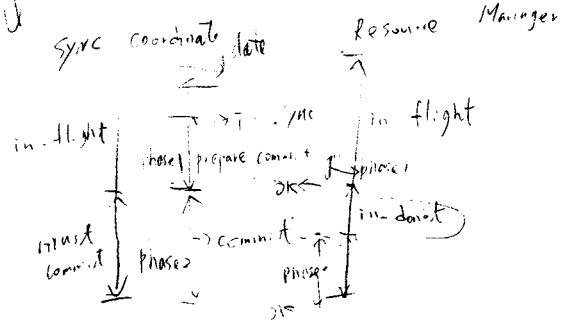
IMS Attachment Facility

Provides support for ...

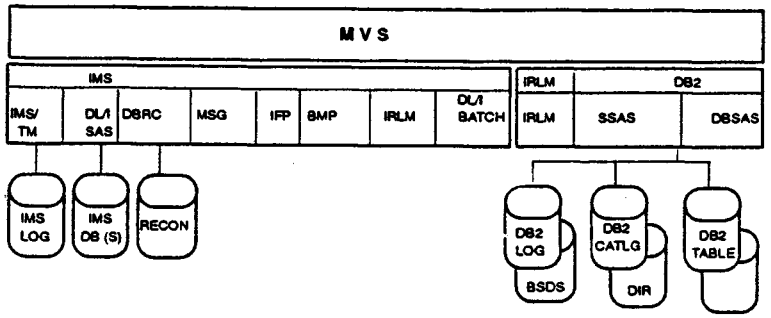
- *Connection* between IMS Subsystems and one or more DB2 Subsystems
- *Communication* between IMS Users and DB2
  - Command Thread
  - Transaction Threads
- Application Programming Interface
  - DB2 Precompiler
  - Language Interface
- Coordinated Recovery/Restart
  - Unit of Work/Unit of Recovery
  - Two-Phase Commit Protocol
  - Indoubt Thread Resolution

Figure 10-9. IMS Attachment Facility (CM011A13)

Notes:



MVS Environment



- IMS and DB2 must be on *same MVS*
- One DB2 S/S can be connected to *multiple IMSs*
- One IMS S/S can connect to *multiple DB2s*
- One IMS dependent region can connect to *multiple DB2s*
- One application program can access *only one DB2* per schedule

Figure 10-10. MVS Environment (CM011A15)

Notes:

IMS DB2-Related Commands

- Subsystem Commands:
  - /DISPLAY SUBSYS xxxx
    displays status of connections
  - /STOP SUBSYS xxxx
    disconnects DB2 from IMS
  - /START SUBSYS xxxx
    reconnects DB2 to IMS after /STOP

- Subsystem Request Command:
  - /SSR -DISPLAY ....
    DB2 Command "DISPLAY ...." sent to DB2

Figure 10-11. IMS DB2-Related Commands (CM011A17)

Notes:

IMS-DB2 System Definition

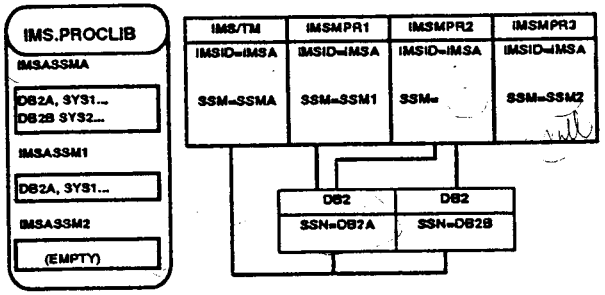
- No specific IMSGEN or DB2 Installation *requirements*, except defining Programs and Transactions to IMS
  APPLCTN PSB=PRGMA
  TRANSACT CODE=TRANA
- Must Define the IMS-DB2 Connections
  Subsystem Member (SSM) in IMS.PROCLIB

- DB2 Pre-compiler:
  - Checks syntax of SQL calls
  - Replaces EXEC SQL with CALL DSNHLI
  - Creates DB Request Module (DBRM)
  - Includes *timestamp* in source and DBRM
- Compile "Modified" Source
  CALL DSNHLI ....
- Link-Edit with IMS call interface module (DFSLI000)
  Entry point is DSNHLI

Figure 10-12. IMS-DB2 System Definition (CM011A19)

Notes:

DB2 Subsystem Member (SSM)

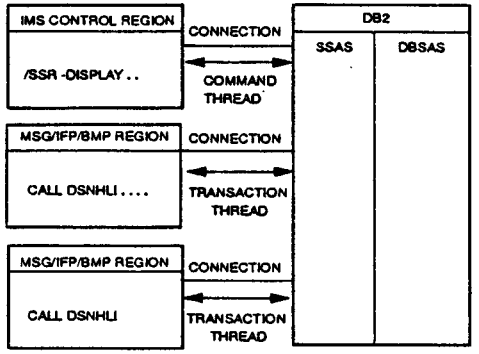


- Subsystem Member identifies IMS/DB2 connections
- SSM Name = IMSID + Suffix (e.g IMSSASSMA)
- Control Region SSM
  - Defines all DB2 connections
- Dependent Region SSM:
  - Defines dependent region connections
  - NULL - defaults to CTL SSM
  - EMPTY - no DB2 connections

Figure 10-13. DB2 Subsystem Member (SSM) (CM011A21)

Notes:

Connections and Threads

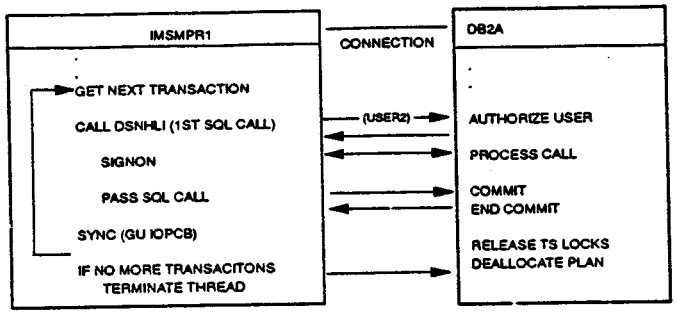


- A *CONNECTION* is an MVS facility that allows subsystems to communicate.
- A *THREAD* is the subsystem control block structure used to communicate with connected subsystems.
- Connection initiated by IMS (IDENTIFY)
  Authorized by RACF
- If DB2x not ready ...
  IMS retries when DB2x available

Figure 10-14. Connections and Threads (CM011A23)

Notes:

IMS-DB2 Transaction Threads



- Thread Reuse:
  - Thread created for duration of PSB Schedule
  - If PROCLIM > 1, multiple transactions per schedule
  - Thread created once for WFI and IFP
- For BMPs, connection made at first SQL call

Figure 10-15. IMS-DB2 Transaction Threads (CM011A25)

Notes:

Security

- RACF Data Set Protection
  - Databases, Libraries, etc.
- RACF Subsystem Connection Verification
  - IMS authorization to connect to DB2
- User Authorization (RACF/IMS/DB2)
  - User signs on to IMS via /SIGN command
- Transaction Authorization (RACF/IMS)
  - User access to IMS transaction
  - User signon to DB2 (plan)
- Command Authorization (IMS/DB2)
  - User authorization to submit IMS and DB2 commands
  - /SSR -DISPLAY

Figure 10-16. Security (CM011A27)

Notes:

10.3 IMS-DB2 Coexistence

Data Propagation: The Problem

- A common question asked by IMS customers is:
  - "How can we take advantage of relational technology in those cases where we need to access data which currently resides in DL/I databases, and where it is not sufficient or acceptable to periodically copy the DL/I database to DB2 tables, or to convert our existing application programs?"
- Both IMS and DB2 offer advantages:
  - Performance
  - Availability
  - Flexibility
  - Ease of use
  - Productivity
  - "Ad hoc" query
- Which DBMS best fits application requirements?

Figure 10-17. Data Propagation: The Problem (CM011A31)

Notes:

Data Propagation: The Solution

- DATA PROPAGATION can provide this support by maintaining two physical copies (hierarchic and relational) as a single logical copy of data, and reflecting changes from DL/I databases into relational databases.
- Systems support for:
  - Data coexistence
    - Decision support system
    - Access by relational query managers
    - Reduce application backlog
  - IMS-to-DB2 migration
    - New applications
    - Conversion of old applications
- IBM Data Propagator MVS/ESA provides:
  - Integrity
  - Consistency
  - Reduce constraints on DBMS choice
  - Select best DBMS for application
  - Protect investments in IMS

Figure 10-18. Data Propagation: The Solution (CM011A33)

Notes:

Date Propagation Model

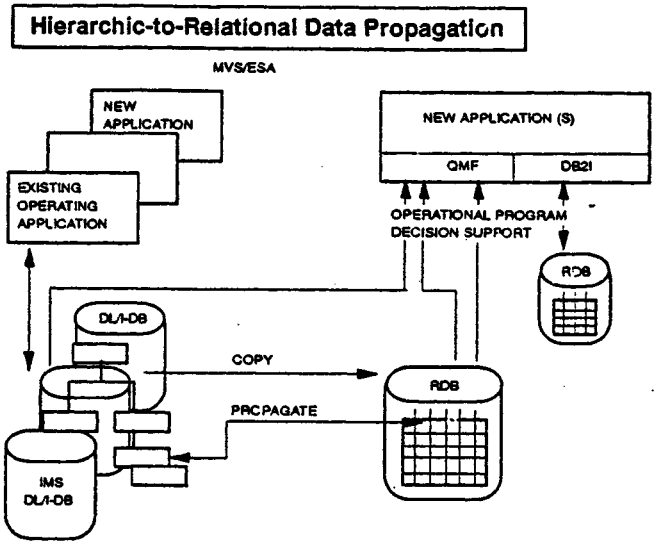


Figure 10-19. Data Propagation Model (CM011A35)

Notes:

DPROP Summary

- IBM Data Propagator MVS/ESA Release 2:
  - Application transparency
  - Definition at DBD-level
  - Flexibility
  - Data integrity
  - Synchronous or Asynchronous
  - Forward and reverse propagation
  - Consistency checking
  - Execution environments:
    - IMS/TM provides:
      - Data Capture Exit
    - IMS/Batch
- Not provided:
  - CICS environments

Figure 10-20. DPROP Summary (CM011A37)

Notes:

Summary

Choice of hierarchical (IMS) or relational (DB2) technology.

Data Propagator to keep data in synch across access methods.

IMS Transaction Manager to allow online access to DB2 tables.

Figure 10-21. Summary (CM011a39)

Notes:

IMS Fundamentals
(Course Code CM01)

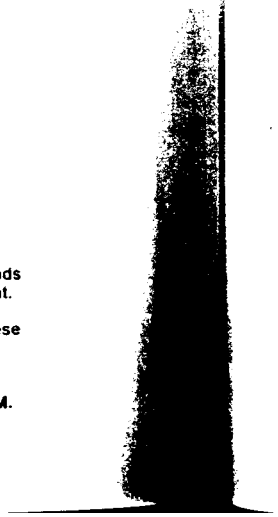
Student Exercises
ERC 1.

IBM Education and Training

Publishing Information

This publication has been produced using BookMaster (Program Number 5668-015), the Document Composition Facility (Program.Number 5748-XX9), and Freelance 2.1 for Windows. It was printed on the IBM 3820 Page Printer.

First Edition (December, 1996)



Contents

Overview Exercise Description

Student-ONLY Overview Exercise Description goes here. These notes will appear in the student exercise guide ONLY.

Exercise 1. IMS / Batch Exercise

What This Exercise is About

- The control blocks (DBD, PSB, and ACB) for a simple batch system, a COBOL application program which uses this mix of control blocks, and a BTS execution of the program.

What You Should Be Able to Do

At the end of the lab, you should be able to:

Identify the major parts of the control blocks and programs

Introduction

This is not a test. The more you discuss this among yourself, the more you will get out of the exercise.

Required Materials

Handout and notes from yesterday's lecture

Exercise Instructions

Analyzing DBDGEN:

1. What is the hierarchic access method chosen.?

2. How many segments are defined ?

3. Quickly sketch the associated hierarchy below:

4. What is the purpose of the FIELD called SKCLASS ?

Analyzing PSBGEN:

5. How are PCBs normally identified in a PSB ?

6. How many PCBs are defined in this PSB ?

7. The program is "sensitive" to which segments in the SKILL database ?

8. Which segments are missing, and why ?

COBOL Program Questions

Analyzing Compiler Output

9. What is the purpose of the LINKAGE SECTION statement (136) ?

10. What is the function of the ENTRY statement (182) ?

11. What is the purpose of the first DL/I call statement (266) ?

12. How is the program terminated ?

Exercise 2. IMS / Online Exercise

What This Exercise is About

The PSB for a simple Message Processing Program, a COBOL MPP itself, and a BTS execution of the program.

What You Should Be Able to Do

At the end of the lab, you should be able to:

Identify the major parts of the PSB and program.

Introduction

This is not a test. The more you discuss this among yourself, the more you will get out of the exercise.

Required Materials

Handout and notes from yesterday's lecture

Exercise Instructions

PSBGEN Process Questions

Analyzing PSBGEN:

1. How are PCBs normally identified in a PSB ?

2. How many PCBs are Defined in this PSB ?

3. What is different about this PSB compared with an IMS/Batch PSB ?

   s (type = TP)'.

4. There is a "silent" PCB that is not indicated. What is it, and why ?

COBOL Program Questions

Analyzing Compiler Output

5. How has the LINKAGE SECTION beginning at (153) been re-structured ?

   s'.

6. How many PCBs are identified in the ENTRY statement (199) ?

7. The first DL/I call statement (216) now does what ?

8. How are the SSA values for the DL/I call statement (246) determined?