IMS/VS Version 1
General Information Manual

Program Product
Program Number 5740-XX2
Release 3

IBM

File No. S370-50   GH20-1260-12

IMS/VS Version 1
General Information Manual

Program Product
Program Number 5740-XX2
Release 3

IBM

File No. S370-50 GH20-1260-12

Thirteenth Edition (February 1984)

This is a major revision of, and makes obsolete, GH20-1260-11.

This edition applies to Version 1 Release 3 of IMS/VS, Program Product 5740-XX2, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are detailed in Appendix A. Because the technical changes in this edition are extensive and difficult to localize, they are not marked by vertical bars in the left margin.

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

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

About This Book

This book introduces Information Management System/Virtual Storage (IMS/VS), a data base/data communications system. IMS/VS runs as a subsystem of either OS/VS1 or OS/VS2 MVS. However, Fast Path and Block-Level Data Sharing are available only with OS/VS2 MVS.

The purpose of this book is to enable managers, system programmers, data base support personnel, and other interested people to evaluate IMS/VS for use in their organization. The book explains the uses of IMS/VS and benefits that organizations can expect when they install the product. It explains:

- What IMS/VS is
- What benefits it provides for the data base system user
- What benefits it provides for the online system user
- The function of IMS/VS's components
- What hardware and software are required to use the product
- What tasks must be performed to install, administer and maintain the product

Organization of the Book

The first chapter of this book is an overview of the entire IMS/VS Data Base/Data Communications System. Subsequent chapters explain each of the major IMS/VS functions. Descriptions of each chapter follow.

Chapter 1, "Introduction,"describes IMS/VS and its advantages for you when you install it. It is intended as an introduction for all readers.

Chapter 2, "IMS/VS Data Base Concepts,"describes the basic concepts readers need to understand about IMS/VS's data base manager, DL/I.

Chapter 3, "IMS/VS Data Communication," describes the Data Communications feature of IMS/VS, which provides network and terminal management for IMS/VS.

Chapter 4, "IMS/VS Application Programming," describes how users write application programs that will operate in the IMS/VS environment.

Chapter 5, "System Control Facilities,"discusses IMS/VS system definition, recovery and restart, and performance and tuning.

Chapter 6, "Using IMS/VS with Other Subsystems,"gives an overview of the use of IMS/VS with CICS/VS, and Database 2 (DB2) subsystems.

Chapter 7, "Tasks in Using IMS/VS," describes installation and maintenance steps.

Chapter 8, "IMS/VS Publications," tells where to find further information on IMS/VS.

Appendix A, "Highlights of IMS/VS Releases,"provides a history of IMS/VS by describing the changes made to the product, in most current to least current order.

Appendix B, "System Configuration and System Control Programs," lists IMS/VS's hardware and software requirements.

Appendix C, "Terminals Supported by IMS/VS,"lists the terminals IMS/VS supports.

Using this Book and the IMS/VS Library

Depending on your reason for reading this book, you may want to give different chapters different amounts of attention. The first chapter provides a high-level overview for all readers. Chapters 2 through 4 provide a slightly more technical view of the Data Base and Data Communication components and the way in which application programming is performed under IMS/VS. Chapter 5 is intended for a technical audience and addresses those components of IMS/VS that are the province of your installation's system programming staff.

Finally, Chapters 7 and 8 provide the information you need to understand the tasks to be performed to use IMS/VS and the books that support these tasks.

Summary of Amendments

Changes to this release of IMS/VS and to previous releases are summarized in Appendix A, "Highlights of IMS/VS Releases."

Contents

Figures

Chapter 1. Introduction

Information Management System/Virtual Storage (IMS/VS) is a proven, reliable data base/data communication (DB/DC) system, capable of managing complex data bases and terminal networks. It can help to make accurate, consistent, current, and often critical information available to many end users in a timely fashion.

The Data Base System

At the heart of IMS/VS are its data bases and its data base manager, Data Language/I (DL/I). DL/I provides the means to create and access IMS/VS data bases. DL/I lets you define the structure of the data base and the relationship of the elements within it. It lets you add new information to your data base, delete unwanted information from it, and update information that already exists within it. Additionally, DL/I permits you to adapt IMS/VS data bases to the requirements of your many and varied applications. It permits application programs to access common and, therefore, consistent data, and reduces the need to maintain the same data in multiple ways in separate files for different applications.

A data base is a collection of interrelated data items organized in a form that may be processed by application programs. IMS/VS data bases are hierarchic. That is, data within the data base is arranged in pyramidal fashion, with data at each level of the hierarchy related to, and in some way dependent upon, data at the preceding level of the hierarchy. Additionally, a specific data item need be stored within the data base just once. It is then available to any user who is authorized to use it. Users do not need to have personal copies of the data.

Figure 1 on page 2 shows a portion of an inventory data base for a retail store. What is important to note is that users from several areas of the retail store's operation can request information from the data base. Each user will see only that information needed; the rest of the information is secure from unauthorized access.

This Introduction will not explain the data base concept further; that discussion is in Chapter 2, "IMS/VS Data Base Concepts" on page 7, later in this *General Information Manual.* Rather, the concept is introduced here in order to stress the advantages provided by IMS/VS data base support:

- Because the same data may be shared by many applications supporting many functions within your company, the data each department receives is consistent.

- DL/I permits application programs to be independent of the physical storage and sequence of the data within the data base. Thus, your investment in data and applications is preserved, since changes to the data base do not require applications to be reprogrammed.

- Redundant data is reduced, as is the effort required to maintain the data.

- Data security is enhanced because of your ability to define and control users' and application programs' access to data base information

Because of these advantages, DL/I can reduce the cost of application development, data processing, and data storage.

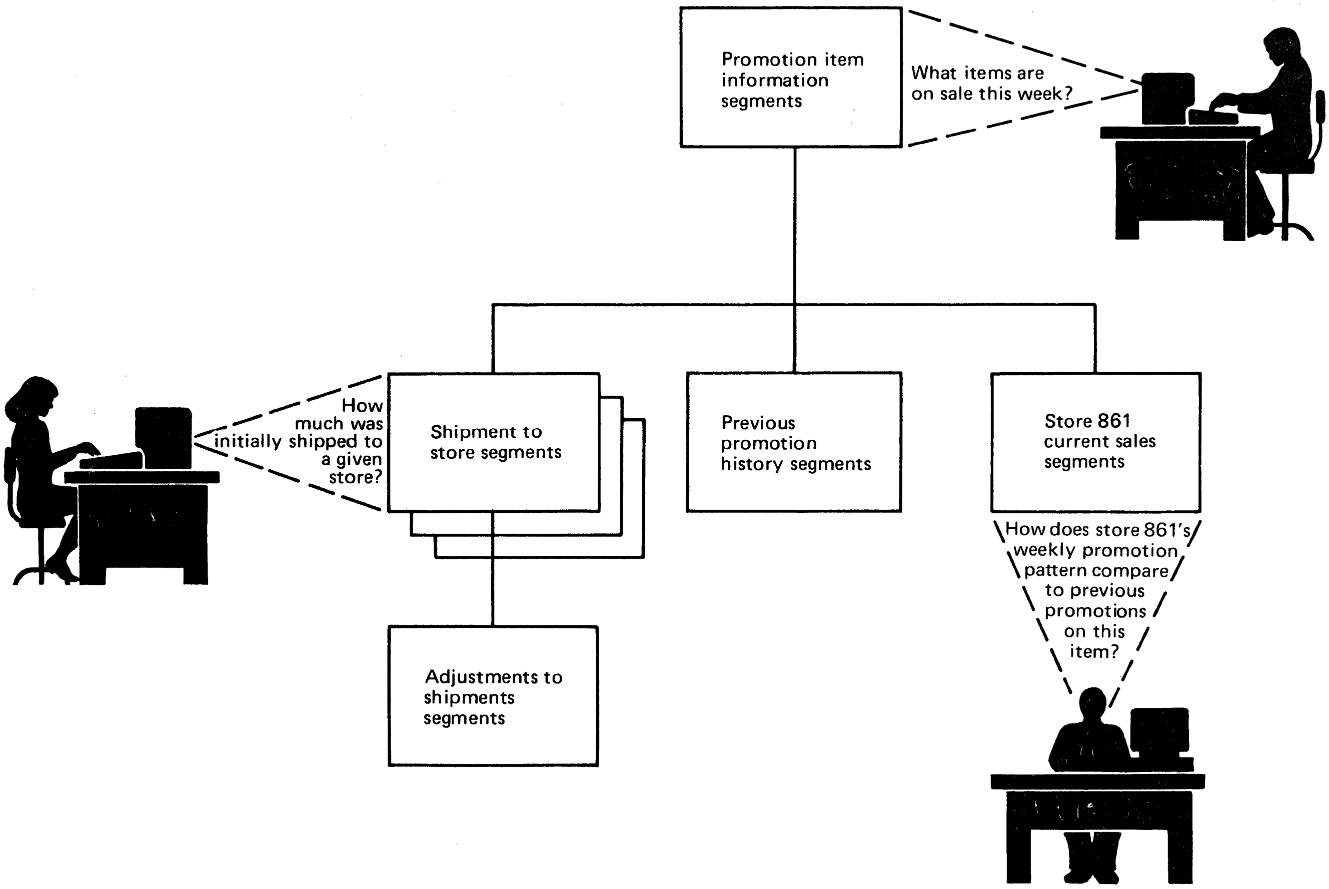

Promotion item information segments
What items are on sale this week?
How much was initially shipped to a given store?
Shipment to store segments
Previous promotion history segments
Store 861 current sales segments
How does store 861's weekly promotion pattern compare to previous promotions on this item?
Adjustments to shipments segments

Figure 1. Current, Consistent Data Is Available to Authorized Users

The IMS/VS Data Base system also provides some additional facilities that increase the flexibility of your IMS/VS system. These are:

- Data Sharing

  Data sharing permits IMS/VS application programs, both batch and online, to access IMS/VS data bases concurrently.

- Data Base Recovery Control

  Data Base Recovery Control (DBRC) assists in recovering IMS/VS DL/I data bases in the event of a failure. DBRC also supports the sharing of data among IMS/VS applications in the same or different processors.

  The log control facility of DBRC manages the online logs and the log archiving process, and records log activity without requiring data bases to be identified to DBRC.

The Data Communication Feature

When the IMS/VS Data Communication (DC) Feature is added to the IMS/VS Data Base System, it extends the facilities of DL/I to the online, real-time environment. It permits users to sit at terminals and enter transactions that have been defined to IMS/VS. These transactions initiate application programs that access IMS/VS data bases and return results.

You may define a variety of online processing options. For example, you may define transactions for high-volume data entry applications, others for interactive applications, and still others to support predefined inquiries. IMS/VS Data Communication supports a wide variety of terminals and devices, as described in Appendix C. It also permits you to develop a wide range of high-volume, rapid-response applications, and to geographically disperse your data processing locations, while keeping centralized control of your data base.

The Data Communication Feature provides some additional components that help you to balance your workload and improve your system's throughput and your staff's productivity. These include:

- Message Format Service

  Message Format Service (MFS) is available for certain terminal types. It simplifies the development and maintenance of terminal-oriented application programs by performing many common application program functions and providing a high level of device independence for the application program.

- Multiple Systems Coupling

  Multiple Systems Coupling (MSC) permits you to link multiple IMS/VS systems and to distribute processing loads and data bases among them to satisfy particular geographic and business requirements. Transactions entered in one IMS/VS system may be passed to another IMS/VS system for processing and the results returned to the initiating terminal. Terminal operators are unaware of these activities; their view of the processing is the same as that presented by interaction with a single system.

- Intersystem Communication

  Like MSC, Intersystem Communication (ISC) permits you to exchange data between multiple IMS/VS systems. In addition, ISC permits communication between IMS/VS and CICS/VS or a user-written subsystem, provided all the subsystems involved have implemented Intersystem Communication protocols. ISC uses System Network Architecture (SNA) protocols. When IMS/VS and another system are connected in this way, both IMS/VS and the other system are defined as SNA logical unit type 6 (LU6) nodes.

- Fast Path

  Fast Path provides data base and telecommunication facilities for applications requiring high transaction rates and high data availability. Operating in existing IMS/VS telecommunication networks, Fast Path provides a message-handling facility to expedite the processing of Fast Path messages.

A further discussion of IMS/VS's Data Communication Feature is found later, in Chapter 3, "IMS/VS Data Communication" on page 23.

IMS/VS Data Communication offers you:

- A way to improve customer and internal service by making current and consistent information available to multiple terminal users
- A way to improve productivity by using terminals for online inquiry and for update applications
- A way to migrate into the online environment

IMS/VS Application Programs

IMS/VS supports several types of application programs:

- Online message processing programs (MPPs)
- Message-driven programs
- Batch message processing programs (BMPs)
- Batch programs

Three major kinds of IMS/VS application programs run in a real-time environment—message processing programs (MPPs), batch message processing programs (BMPs), and message-driven programs. Because these programs use IMS/VS resources concurrently, IMS/VS monitors them, and schedules and controls their use of system resources such as message queues and data bases.

MPPs are IMS/VS application programs that process messages from and send replies to IMS/VS message queues. They are used only for message processing.

Message-driven programs are similar to MPPs: Their main purpose is to quickly process and reply to messages from terminals. Unlike MPPs, message-driven programs bypass IMS/VS scheduling, allowing for more efficient processing; transactions are processed by Fast Path's expedited message handling.

Batch message processing programs (BMPs) are IMS/VS application programs that can perform message processing or batch-type processing online. Like MPPs, BMPs are run online, but are started by job control language. (In contrast, MPPs are invoked through transaction codes that are entered at a terminal.) Unlike MPPs, BMPs can access operating system files, in addition to IMS/VS data bases. You can find more information about IMS/VS application programs later in this book in Chapter 4, "IMS/VS Application Programming" on page 33.

Batch programs accept accumulated input and periodically process it against the data base. Because the data base is not continuously available to the batch processing user, the information in the data base may not be completely up-to-the-minute. However, the user of batch applications may not always require the most current information. Application programs that produce a large volume of reports are usually run as batch programs.

IMS/VS permits data to be shared by batch and online application programs. IMS/VS application programs located in the same or separate IMS/VS systems and operating in the same or separate processors can access data bases concurrently. IMS/VS controls access to the data and assists in maintaining its integrity. IMS/VS also provides a means of preventing unauthorized access to data. Careful definition of an application program's processing prevents an application from accessing any portion of a data base but that portion it has been given authority to access. Thus, data can be secured from unauthorized access and update.

Application programs may be written in COBOL, PL/I, or assembler language. Interspersed within the application program are DL/I statements or "calls," which the application program uses to process the data base.

IMS/VS application programming support provides you with the following advantages:

- IMS/VS permits the evolutionary expansion of your applications from the batch environment to the online environment. An application can initially use DL/I for batch-only applications. Once experience is gained and the needs of the business dictate, the same data base and application program design may be used in the online environment.

- Application programs are independent of the physical storage organization of the data base. Even if a data base is restructured or expanded, you may not have to alter the application programs that use it.

- Application programs may access data items within a data base either sequentially or directly.

- Batch and online application programs may operate concurrently within IMS/VS.

- Data may be concurrently shared by application programs.

IMS/VS Support for Recovery and Security

IMS/VS is a recoverable system. Programs and procedures are provided that enable IMS/VS to recover from failures, often with little or no human intervention. This built-in recoverability helps maintain the integrity of your data base across system failures, and ensures that your personnel have the means for recovery, if it is necessary.

Additionally, IMS/VS provides a way of ensuring system security. Data and other IMS/VS resources can be protected from unauthorized access and alteration.

Both recovery and security are discussed in more detail in Chapter 5, "System Control Facilities" on page 39.

The major advantages of IMS/VS support include:

- The ability to recover from failures quickly, using system-provided tools

- The ability to secure your system and its resources from unauthorized access and change

IMS/VS Advantages

To summarize, IMS/VS offers you a powerful system that:

- Provides a clear migration for your installation from batch processing to online, real-time processing

- Provides centralized control of your data base, while supporting geographically distributed processing centers to make processing results available when and where needed

- Protects your investment in data and applications by minimizing and controlling the effects of change on existing data bases and application programs

- Assists in maintaining the integrity of your data base

- Shields your end-users from the effects of alterations to the system and its data

- Provides recovery in the event of a failure, often with little or no human intervention

- Provides a way to secure system resources from unauthorized access and alteration

Chapter 2. IMS/VS Data Base Concepts

DL/I provides comprehensive data base management, while requiring minimal system and application programming effort. Highlights of DL/I are that it:

- Reduces application program maintenance. With DL/I, minimal changes to applications are required when record layouts change, and you can add and delete applications with little disruption to operations.

- Permits data to be added to or deleted from the data base with little effect on application programs that use existing data.

- Reduces data redundancy. With DL/I, you only have to store information once. This saves storage space and cuts down on the time required to process updates.

- Reduces data maintenance. Because you only have to store data once, you have to make changes only to this one copy of the data.

- Provides a comprehensive set of utility programs that assist in maintaining data integrity during system restarts and during the recovery process.

- Minimizes device and access method dependencies. With DL/I, storage devices and access methods can be changed with little or no impact on application programs.

- Provides security. DL/I allows only authorized personnel to access the data base.

- Increases overall control over data. With DL/I, information is centralized as much as possible, which increases data control.

To understand how DL/I achieves these objectives, it is necessary to look at DL/I in more detail. The following sections describe:

- Data organization

- Data bases and access methods

- Optional functions

- How a data base is defined to IMS/VS

- How application programs view the data base

- Additional IMS/VS Data Base facilities

  - Data Base Recovery Control

  - Data Sharing

- Aids for data base installation and maintenance

Data Organization

The data in a data base is grouped into a series of **data base records**. Each data base record is composed of smaller groups of data called **segments**. Generally, a segment is the smallest piece of data DL/I can store; it is the unit of data transferred between an application program and DL/I. Segments, in turn, are made up of one or more **fields**.

Figure 2 shows a data base record in a data base for a school. Each of the boxes in Figure 2 is a segment. The segments in the data base record contain the following information:

- COURSE—the name of the course
- INSTR—the name of the teacher of the course
- REPORT—a report the teacher wants at the end of the course
- STUDENT—the names of students in the course
- GRADE—the grade a student got in the course
- PLACE—the room in which the course is taught

The segments within a data base record exist in a pyramid-like order called a **hierarchy**. The COURSE segment is at the top of the hierarchy, and is called the **root segment**. (There can be only one root segment in a data base record.) All other segments in the data base record (INSTR, REPORT, etc.) are called **dependent segments**. Their existence depends on there being a root segment. Without the root segment COURSE, there'd be no reason for having, for example, a PLACE segment saying which room the course is to be held in.

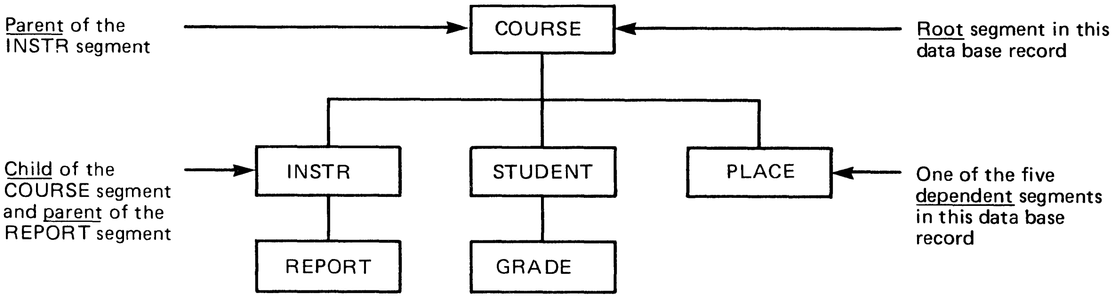

Parent of the INSTR segment
COURSE
Root segment in this data base record
Child of the COURSE segment and parent of the REPORT segment
INSTR
STUDENT
PLACE
One of the five dependent segments in this data base record
REPORT
GRADE

Figure 2. A Data Base Record in a School Data Base

There's another set of words used to refer to how segments relate to each other in a hierarchy: parent and child. A **parent segment** is any segment with a dependent segment beneath it in the hierarchy. COURSE is the parent of INSTR. INSTR is the parent of REPORT. A **child segment** is any segment dependent on another segment above it in the hierarchy. REPORT is the child of INSTR, and INSTR is the child of COURSE. INSTR is a parent segment in its relationship to REPORT and a child segment in its relationship to COURSE.

Two other terms are used for describing segments: type and occurrence. **Segment type** and **segment occurrence** are used to distinguish between a **type** of segment in the data base (for example, the COURSE segment) and a **specific** segment (for example, the COURSE segment for an ALGEBRA course). Figure 2 is a **design** for a data base record. It shows the segment types that will be in the data base.

Figure 3 shows an actual data base record based on this design. Because each segment is a specific one (BAKER and COE, not STUDENT), each segment is a segment occurrence. A segment occurrence is a single specific segment. Within most data bases, there can be 255 segment types; the number of segment occurrences is limited only by the amount of storage allocated for the data base.

There's one other term for describing segments that is important: twin. **Twin segments** are multiple occurrences of the same segment type under a single parent. In Figure 3, BAKER and COE are twins. They have the same parent (ALGEBRA), and BAKER and COE are the same segment type (STUDENT).

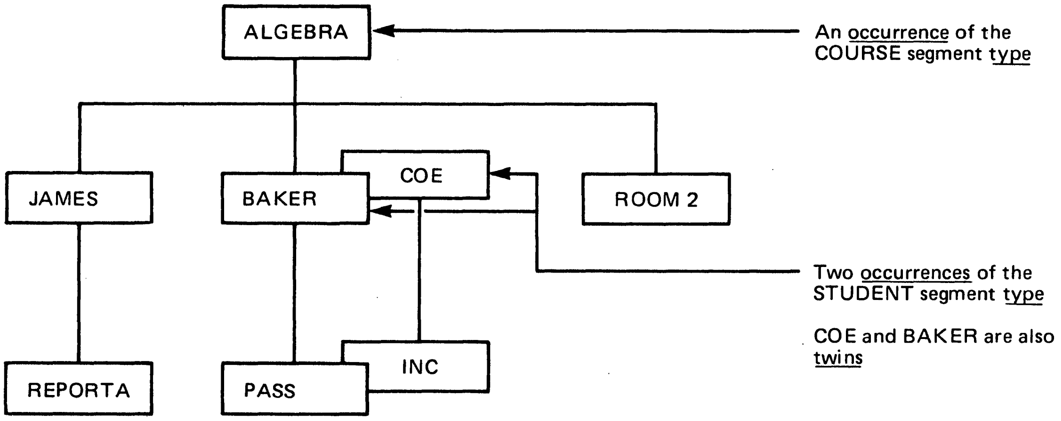

ALGEBRA
An occurrence of the COURSE segment type
JAMES
BAKER
COE
ROOM 2
Two occurrences of the STUDENT segment type
COE and BAKER are also twins
REPORTA
PASS
INC

Figure 3. The School Data Base Record

When a data base is initially created, data base records are stored in it starting at the top of a data base record (at the root segment), in the sequence shown by the numbers in Figure 4 on page 10. This sequence in which segments are stored is called "top to bottom, left to right."

DL/I allows an application program to retrieve segments sequentially or directly. With sequential access, segments are retrieved in hierarchic sequence. Segments may also be retrieved directly without going through the sequential processing sequence.

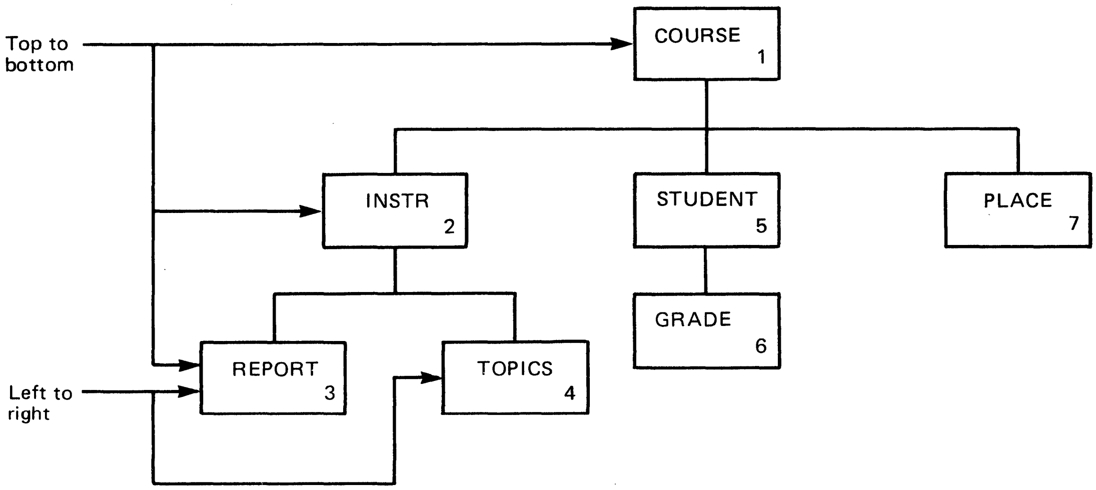


Figure 4. Sequence in a Hierarchy

Data Bases and Access Methods

IMS/VS allows nine different types of data bases to be defined. These types of data bases each meet different application processing requirements. For each of your data bases, the data base type your installation defines should be the one that best satisfies its processing requirements. Basically, data bases are organized either sequentially or directly. With both methods of organization, data may be indexed. Some of the types of data bases serve a special purpose, for example, allowing existing programs to remain usable when an installation is converting to IMS/VS.

Figure 5 lists the types of IMS/VS data bases that can be defined, the IMS/VS access methods they use, and the operating system or IMS/VS access methods that can be used in conjunction with them.

Type of Data Base	IMS/VS Access Method Used	Operating System Access Methods that Can Be Used
HSAM	Hierarchic Sequential Access Method	BSAM or QSAM
SHSAM	Simple Hierarchic Sequential Access Method	BSAM or QSAM
HISAM	Hierarchic Indexed Sequential Access Method	VSAM or ISAM/OSAM[1]
SHISAM	Simple Hierarchic Indexed Sequential Access Method	VSAM
GSAM	Generalized Sequential Access Method	BSAM or VSAM
HDAM	Hierarchic Direct Access Method	VSAM or OSAM[1]
HIDAM	Hierarchic Indexed Direct Access Method	VSAM or OSAM[1]
MSDB[2]	Main Storage Data Base	None
DEDB[2]	Data Entry Data Base	VSAM

Figure 5. Types of IMS/VS Data Bases and the Access Methods They Can Use

Notes to Figure 5:

[1] OSAM is an access method available only in IMS/VS.

[2] These Fast Path data bases are described in Chapter 3, "IMS/VS Data Communication" on page 23.

Optional Functions

The following are optional data base functions:

Name	Function
Logical relationships	Accessing segments from multiple data bases in different sequences
Secondary indexing	Providing alternate paths to data
Variable-length segments	Accommodating variable-length data
Segment edit/compression	Editing and compressing data
Field level sensitivity	Changing field sequence and restricting access
Multiple data set groups	Separating segments for improved performance
Data partitioning	Ability to specify multiple areas
Data replication	Ability to make multiple copies

Logical Relationships

Logical relationships is a function that lets an application program access a logical data base record. A logical data base record can consist of segments from one or more physical data bases. So a logical data base record lets an application program view a data base structure that's different from the physical data base structure.

An advantage of using logical relationships is that programs can access the data as though it existed in more than one hierarchy.

If, for example, a logical data structure is defined that contains segments from two different physical data bases, a segment can be accessed from two different paths:

- A segment can be physically stored in the path where it's most frequently used and where the most urgent response time is required.

- A pointer containing the location of the segment can be physically stored in the alternate path needed by another application program.

If you don't use logical relationships and if two application programs need to access the same data through different paths, you would have to store the data in both hierarchies.

Secondary Indexing

Secondary indexing allows you to access data base records in a sequence other than that defined by the root key. If you don't use secondary indexing, accessing a segment qualified by other than the key can be inefficient; DL/I must search several segments to find the correct one. With secondary indexing, DL/I can directly access a segment based on a field value that is not the key field.

Variable-Length Segments

Variable-length segments is a function that allows the data portion of a segment type to be variable in length. You'd want to use variable-length segments when the size of the data portion of a segment type varies greatly from one segment occurrence to the next. With variable-length segments, you define the minimum and maximum length of a segment type. This saves space in the data base whenever a segment is shorter than the maximum length.

Field Level Sensitivity

Field level sensitivity is a function used to:

- Allow an application program to use a subset of the fields that make up a segment (so it doesn't have to process fields it doesn't use) or to use fields in a segment in a different order. You'd use field level sensitivity in this way to accommodate the differing needs of application programs.

- Allow an application program to access only selected fields in a segment (for security purposes).

- Add fields to a segment without affecting the way in which current applications process fields that now exist in that segment.

Segment Edit/Compression

Segment edit/compression is a function that provides an exit to permit you to:

- Encode or "scramble" segment data when it's on the device so that only application programs with access to the segment can be given the data in decoded form.

- Edit data so that application programs can receive data in a format other than the one in which it's stored.

- Compress data when a segment is written to the device so that DASD storage is better utilized.

Multiple Data Set Groups

Multiple data set groups is a function that allows some of the segments in a data base record to be put in data sets other than the primary data set. This can be done without destroying the hierarchic sequence of segments in a data base record. This permits you to separate seldom-used data base segments from those used often. Thus, an application program can quickly access the segments it's interested in and bypass the data sets containing segments it's not interested in.

Another reason for using multiple data set groups is to improve performance, for example, by separating high-use segments from low-use segments. Or multiple data set groups might be used to save space by putting in a separate data set group those segment types whose size varies greatly from the average.

Data Partitioning

Data partitioning allows you to divide a data entry data base (DEDB) into several areas. Each area contains a different collection of data base records, grouped together by a randomizing algorithm. The area concept makes it possible for

DEDBs to provide a high level of data availability and to support large data bases. Areas are independent of each other; thus, an error in one area does not affect other areas in the DEDB. Areas may be different sizes and stored on different device types. This makes it possible, for example, for you to store your most frequently accessed data in an area stored on a fast device type. Figure 6 shows four areas of a DEDB, each containing a different range of keys.

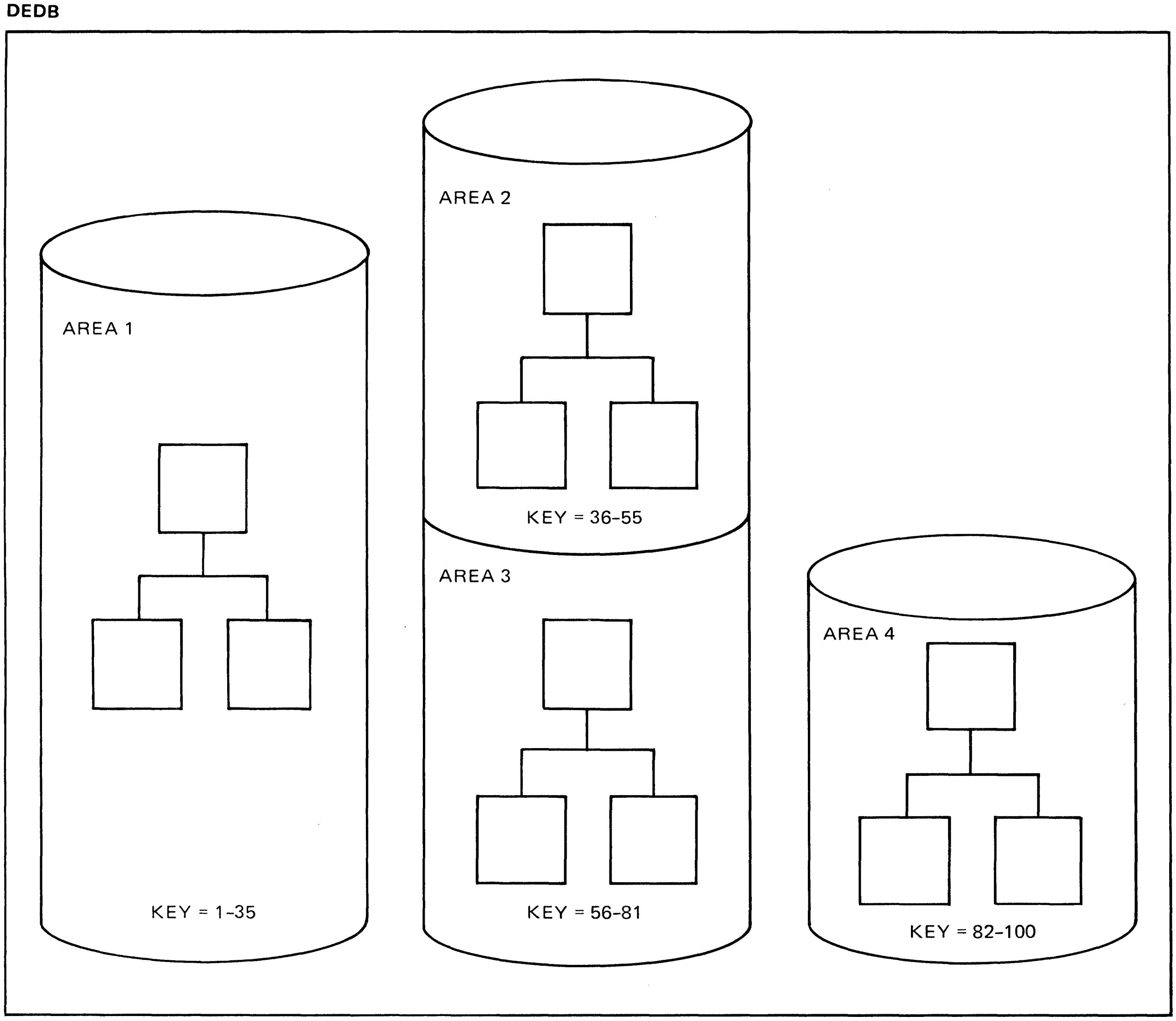


Figure 6. An Example of Areas in a DEDB

Data Replication

Data replication allows you to make as many as seven copies of each area data set in a DEDB, further increasing the high data availability of a DEDB. If there is an error in one area data set and if a copy of that data set exists, application programs may continue processing the data in the copy.

How a Data Base is Defined to IMS/VS

The characteristics of a data base are defined to IMS/VS by coding and generating a data base description (DBD). A DBD describes such things as a data base's organization and access method, the segments and fields in a data base record, and the relationship between types of segments.

Defining the DBD outside the application program achieves both program independence and device independence. This lets you add new storage devices without necessarily having to change your application programs.

If you have the IBM DB/DC Data Dictionary, you can use it to help define your data base. It can contain most of the information needed to produce a DBD.

How Application Programs View the Data Base

The way in which an application program views the data base can be controlled. An application program may not need to use all the segments or fields in a data base record. Or, for security or integrity purposes, you may not want an application program to have access to specific segments or fields. In addition, you may not want an application program to perform certain types of operations on some segments or fields. For example, you might want to give an application program access to a SALARY segment, but not allow it to update the segment. The segments and fields an application can view and the operations it can perform on a segment are controlled by a program specification block (PSB).

A PSB describes an application program's view and use of segments in the data base. An application program may have different views and uses of the same data base. It can also access several different data bases.

If you have the IBM DB/DC Data Dictionary, you can use it to help define an application program's view of the data base. It can contain most of the information needed to produce a PSB.

Additional IMS/VS Data Base Facilities

The sections below describe some additional Data Base system facilities that expand your IMS/VS system's capabilities.

Data Base Recovery Control

Data Base Recovery Control (DBRC), which is included with your IMS/VS Data Base system, permits easier recovery of IMS/VS data bases in your IMS/VS installation.

DBRC helps control and keep track of the following:

- Logging.

  For an online system, DBRC keeps track of the logs available for reuse, and of the logs containing records required for restart. For a batch system, DBRC keeps track of log serial numbers and start and stop times for all logs created.

- Data base recovery.

  DBRC stores information about events that might affect recovery in two recovery control (RECON) data sets. IMS/VS uses the recovery control data set information (such as data set names, volume serial numbers, and the times of creation) to determine what data sets are needed as input during a restart or recovery.

DBRC can control three types of data base recovery:

- Full recovery, which results in the recovery of a data base or data base data set to its condition before it failed
- Time-stamp recovery, which results in the reconstruction of a data base data set to its condition at some preceding time, such as the time at which a log data set was closed or the time at which an image copy was made
- Track recovery, which results in restoring a failed track of a VSAM data base data set

For each type of recovery, one command to the DBRC recovery control utility generates both the JCL and the control statements needed to perform the specified recovery.

- Data sharing

  DBRC supports the sharing of information among IMS/VS applications in the same or different processors. The recovery control data sets hold authorization control information that assists IMS/VS in controlling and serializing application programs' access to and display of information from shared IMS/VS data bases.

You can use a subset of the function provided by DBRC. If you don't use IMS/VS data sharing, you can restrict the use of DBRC to:

- Tracking and controlling IMS/VS logging
- Controlling data base recovery, in addition to logging

Sharing Data among Multiple Subsystems

When several application programs in one IMS/VS system run online, an IMS/VS system (without data sharing) allows them to read and update IMS/VS data bases concurrently. It does this by preventing a program from accessing data that another program is updating until the updating program indicates to IMS/VS that its changes are valid. Data sharing extends what a single IMS/VS system does by making it possible for application programs in separate IMS/VS systems—running in the same or separate processors—to access the same data bases concurrently. Batch and online systems can share IMS/VS data bases. CICS/VS subsystems can participate in data sharing, and CICS/VS online programs can share data bases with IMS/VS online and batch programs. Data sharing is not supported for either MSDBs or GSAM data bases.

Data sharing is a part of the IMS/VS Data Base System. There are two levels at which IMS/VS systems can share a data base: the data base level and the block level.

- When IMS/VS systems share a data base at the **data base level**, application programs in each IMS/VS system can read the data concurrently; or one IMS/VS system can update the data while the other IMS/VS systems have read-only access to it. Read-only access means that application programs are able to read the data, but they are not protected from uncommitted data. At the data base level, IMS/VS systems share the entire data base.

- When IMS/VS systems share a data base at the **block level**, several application programs can update the data concurrently. To the IMS/VS systems accessing the data, it appears that they are reading and updating the data at the same time—that they are literally sharing the data base. In reality, IMS/VS serializes the requests and controls access to the data base so that application programs are protected from uncommitted data. When IMS/VS systems share an OSAM or an ISAM/OSAM data base at the block level, IMS/VS serializes their requests to a physical block of the data base. When IMS/VS systems share a VSAM data base, IMS/VS serializes their requests to a control interval (CI) of the data base.

Data Base Recovery Control (DBRC) and the IMS/VS Resource Lock Manager (IRLM) provide additional support for data sharing. All data bases that will be shared must be registered with DBRC. DBRC controls the ability of a given IMS/VS system to access a data base.

If data bases will be shared at the block (or control interval) level, IRLM is required. IRLM services are used by IMS/VS to serialize application program requests for data base records and to ensure that two programs do not access the same record for update at the same time.

To summarize, at the data base level, only one IMS/VS system can update the data; all other systems have read-only access to the data. At the block level, IMS/VS systems can update the data concurrently. The levels of sharing are the same, regardless of whether the IMS/VS systems run in the same processor or in separate processors. Data sharing between IMS/VS systems in the same processor is called **intraprocessor sharing**. Data sharing between IMS/VS systems in separate processors is called **interprocessor sharing**.

Sharing at the Data Base Level

Figure 7 illustrates how IMS/VS systems share data at the data base level. One system is allowed to update the data; access to the shared data base by all other IMS/VS systems—both online and batch—must be read-only. IMS/VS uses Data Base Recovery Control (DBRC) to control access to the data base.

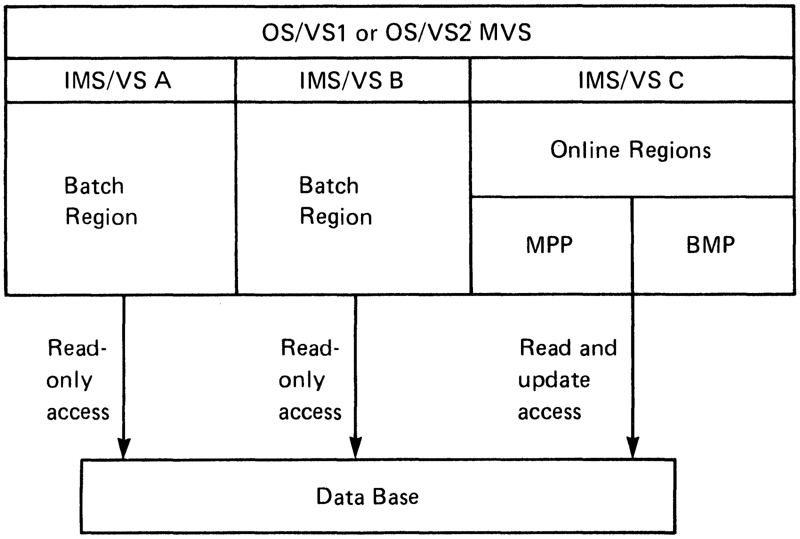


Figure 7. Data Base Level Sharing—Intraprocessor

In Figure 7, the three IMS/VS systems (IMS/VS A, IMS/VS B, and IMS/VS C) reside in the same processor. IMS/VS A is a batch system, as is IMS/VS B; and IMS/VS C is an online system. Both the MPP and the BMP in IMS/VS C can update the data base. The batch programs running in IMS/VS A and IMS/VS B have read-only access to the data base (and aren't protected from reading uncommitted data). The read and update rules for data base sharing would be the same if one or two of the IMS/VS systems were running in a separate processor. At the data base level, the maximum number of processors that can participate is limited by the number of paths to the direct access devices on which the data to be shared resides. Data base level sharing can run under OS/VS1 or OS/VS2 MVS.

Sharing at the Block Level

Sharing at the block level is the same for intraprocessor and interprocessor sharing. Block level sharing can run only under OS/VS2 MVS. For block level sharing, the IMS/VS systems can be running in a maximum of two processors. Multiple IMS/VS systems may update the data concurrently. With block level sharing, the systems reading the data are protected from reading uncommitted data. Figure 8 on page 19 illustrates a possible data sharing configuration.

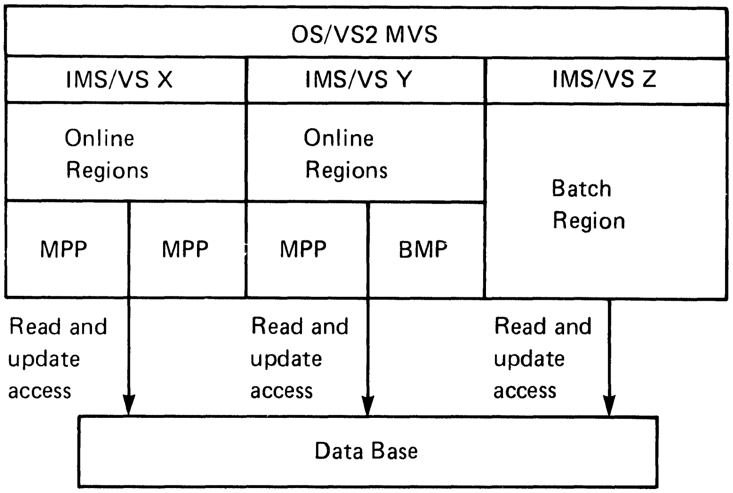


Figure 8. Block Level Sharing with Online and Batch System Update

Data Sharing in Mixed Configurations

It is also possible for IMS/VS systems running under different operating systems to share a data base. For example, suppose IMS/VS systems in two processors are sharing a data base at the block level. To do this, both IMS/VS systems must run under OS/VS2 MVS. IMS/VS systems running in another processor under OS/VS1 can also share the data base, but only at the data base level.

How Data Sharing Is Useful

Data sharing is valuable to IMS/VS installations because it helps them handle increasing IMS/VS processing workloads and application requirements. As the number of applications grows, IMS/VS installations need their IMS/VS systems to be more available and have greater capacity. Some of the areas in which data sharing can be of help are described below:

- Because data bases are accessible to multiple IMS/VS systems in one or more processors, an installation can spread the IMS/VS workload across several processors.

- Because both batch and online systems can concurrently share a data base, an installation can mix its batch and online applications to optimize the performance of its IMS/VS system(s).

- The time required for batch processing can be reduced because installations can run multiple batch readers—programs that create reports and can be long running—at the same time.

- Because data sharing provides a means of controlling data base access, production and test systems can run at the same time instead of separately. This can simplify test operation procedures.

- Data sharing permits batch utilities to run concurrently with production jobs; for example, the Image Copy utility can share data bases with read-only applications.

Aids for Data Base Installation and Maintenance

A full set of IMS/VS utilities and other programs assists in the following areas:

- Data base and program testing
- Data base loading
- Data base monitoring
- Data base maintenance

Data Base and Program Testing

A test data base, usually using information extracted from current files, should be built for preinstallation testing. The IBM Data Extraction, Processing, and Restructuring System (Program Number 5796-PLH) can help in this area by allowing an installation to:

- Combine components of different files, or restructure old files to form new files
- Rearrange data within a file
- Alter values within a file
- Build hierarchies and change the number of levels in a hierarchy

Another utility, DBPROTOTYPE II (Program Number 5796-PJK), assists in designing, loading, and testing data bases. By running both the test and production data base against the production program specification blocks and data base descriptions, reports can be generated to ensure that the test data base is yielding the expected results.

The DL/I Test Program, which is a part of IMS/VS, and two other IBM products, the IMS/VS Space Management Utilities II (SMU II; Program Number 5796-PJJ) and the Application Development Facility (ADF; Program Number 5796-PHX), may also be helpful in creating and testing data bases. ADF allows interaction with the test data base online.

The Batch Terminal Simulator II (BTS II; Program Number 5796-PGT) permits testing of online programs in an IMS/VS batch region, without the use of terminals, by simulating terminal input.

The IBM DB/DC Data Dictionary (Program Number 5740-XXF) is a development and administrative tool that manages information about an installation's data processing resources. The Data Dictionary contains definitions of the elements in the production data bases. The user can determine from these definitions those elements to be included in the test data base and copy them, assigning the copy a test status. Additionally, Data Dictionary can assist in generating the data base descriptions and program specification blocks for the test data base.

Data Base Loading

A user-written load program is required for the initial data base loading. A load program builds sets of segments and issues a series of insert calls to build the data base.

You can also use tools such as the DL/I test program and the Data Extraction, Processing, and Restructuring System (Program Number 5796-PLH) as tools to load your data bases, provided the input data stream is in a form acceptable to these tools.

Data Base Monitoring

The Data Base Monitor (DB monitor) records data about the performance of data bases in a batch environment and produces a variety of reports. The monitor's use is twofold. First, the monitor gives performance data over a period of time. By comparing this data, performance trends can be determined. Second, the user can test the effect of changes to data bases on performance. That is, baseline figures can serve as a profile against which to compare the effect of changes. Thus, it can help indicate when tuning or data base reorganization is needed.

IMS/VS Monitor Summary and System Analysis Program II (Program Number 5798-CHJ) provides DB/DC system performance data and serves as a tuning aid.

Space Management Utilities II (Program Number 5796-PJJ) supplies information on how data is stored and helps determine the best space utilization. These utilities monitor data base growth rates and provide statistics that allow resequencing and reloading of segments during data base reorganization.

DBPROTOTYPE II provides estimates of direct access storage device (DASD) space and the amount of I/O that will be required; also, processor and I/O time estimates for application program calls.

The DL/I STAT call retrieves statistics on data base buffer pools.

In addition to its test function, the Batch Terminal Simulator (above) allows monitoring of the data base, providing data on IMS/VS control blocks and work areas. OS/VS access method utilities also can provide useful information on IMS/VS data base operations.

Data Base Maintenance

The facilities described below assist in maintaining your DL/I data bases.

Data Base Surveyor Utility Feature

The output of this utility can help determine the need to partially reorganize data bases. DB Surveyor produces a report describing the physical organization of the data base, and includes the size and locations of free space. This assists in locating free space into which reorganized data base records can be loaded.

Reorganization Utilities

There are two sets of utilities that enable unloading and reloading of Hierarchic Indexed Sequential Access Method (HISAM) and Hierarchic Direct (HDAM, HIDAM) data bases. A Partial Data Base Reorganization utility also allows reorganizing a portion of a data base.

The DEDB Direct Reorganization utility reduces space fragmentation in DEDBs.

Two utilities, the Data Base Prereorganization and the Data Base Scan, are used for loading and reorganizing data bases with logical relationships. The Data Base Prereorganization utility is also used for data bases with secondary indexes.

Data Base Recovery Control

Data Base Recovery Control (DBRC) assists in maintaining and recovering data bases. DBRC generates most of the JCL and control statements needed to run various data base recovery utilities. It records the occurrences of data base reorganizations or reloads. It also assists in recovering data base data sets and can reduce errors that may occur if manual recovery procedures are used. In addition, DBRC facilitates sharing of data bases among IMS/VS systems and aids in controlling log information.

Chapter 3. IMS/VS Data Communication

The IMS/VS Data Communication (DC) Feature offers you:

- A way to improve your customer and internal service by providing current and consistent information when and where it is needed
- A way to distribute processing function to the locations at which it is needed
- A way to improve productivity by using terminals for online inquiry and update applications

This chapter describes the IMS/VS online system, highlights some of its major processing functions, and discusses how it is controlled. Its contents include:

- Major IMS/VS Data Communication Functions
- Controlling Online IMS/VS Operation
- Additional IMS/VS Data Communication Facilities
  - Fast Path
  - Message Format Service
  - Multiple Systems Coupling and Intersystem Communication

Major IMS/VS Data Communication Functions

The IMS/VS Data Communication Feature:

- Manages the IMS/VS terminal network
- Routes messages from terminal to terminal, from application to application, and between application programs and terminals
- Queues input and output messages
- Schedules application programs
- Provides other system control facilities for system definition, restart and recovery, and performance and tuning

The first four topics are addressed in this chapter. The last topic is addressed in Chapter 5, "System Control Facilities" on page 39.

Managing the Terminal Network

IMS/VS Data Communication supports the attachment of many types of terminals, including display devices, card readers, high speed printers, and remote subsystems. IMS/VS Data Communication uses the network facilities of communications managers such as BTAM and VTAM for online processing. You describe the physical configuration of these devices during system definition (as explained in Chapter 5, "System Control Facilities" on page 39).

In addition to describing the physical network to IMS/VS, you also define **logical terminals**. A logical terminal, or **LTERM**, names a logical device that is associated with a physical terminal. One physical terminal may have one or more logical terminals associated with it. You refer to the logical terminal in the construction and transmission of your input and output messages. Thus, when designing application programs, you are freed of the need to concern yourself with actual terminal addresses or the availability of the device. Additionally, if a physical terminal becomes inoperative, the logical terminal(s) associated with it can be dynamically reassigned to another physical terminal. This permits messages waiting to be sent to the inoperative terminal to go to another physical destination.

The LTERM concept also enhances IMS/VS security, because each logical terminal can have unique security parameters associated with it.

Routing Messages

IMS/VS processes three basic types of input messages. The first one to eight characters of the input message determine the message type and identify the destination of the message text. An input message may be:

- A **message** to be processed by an IMS/VS application program. This type of message is called a **transaction** and the first one- to eight-character identifier is called a **transaction code**.The transaction code determines the application program that is the **destination** of the message.

- A **message switch**. If the first one to eight characters of the input message contain the name of an IMS/VS logical terminal (LTERM), the message is sent directly to that LTERM without the intervention of an application program.

- A **command**. If the first character of the input message is a slash (/), the message directs IMS/VS to perform a function such as displaying or altering the status of one or more IMS/VS system resources. Certain IMS/VS commands that control IMS/VS resources can be restricted to certain classes of users or to the IMS/VS master terminal operator (MTO). (The master terminal operator controls IMS/VS facilities. MTO functions are discussed later in this chapter under "Master Terminal Operator Control of IMS/VS" on page 27. IMS/VS commands can also be entered by IMS/VS application programs using the Automated Operator Facility (AOF, 5740-XYD). These topics are discussed further in the section "Controlling Online IMS/VS Operation."

In addition, IMS/VS also accepts data that continues a conversation, that is entered after an input destination has been established via IMS/VS's preset destination mode facility, or that requests presentation of specific parts of multipage messages handled by IMS/VS's Message Format Service.

Queuing Input and Output Messages

After a message has been entered and its destination is determined, IMS/VS places the message on an input queue for the specified destination. Queuing of an input transaction permits a terminal operator to continue to enter data into the system even though the application program needed to process the transaction may not be immediately available. Thus, a terminal operator may continue to enter new transactions while IMS/VS processes transactions entered earlier.

After an IMS/VS application has completed processing a transaction, it places the output message back on a queue. The output may be queued for the terminal from which the input came, or an alternate destination for the output may be selected when the application program is defined. If the output is being returned to the entering terminal, IMS/VS places it in that terminal's queue, from which the operator can request it when it is needed.

Message queuing has several important advantages:

- When a message that is defined as recoverable is placed on a queue, it can be recovered should the IMS/VS system shut down on either a scheduled or an unscheduled basis.

- It permits IMS/VS to process input independently of the entry of data from a terminal. This frees the terminal operator to continue data entry or to view output results while IMS/VS is processing input from that terminal.

- IMS/VS permits long and short messages to be queued separately on multiple direct access data sets. This has the potential to improve system performance.

For applications that require fast message processing, Fast Path provides an expedited message handling. Messages to and from a message-driven program bypass IMS/VS message queue processing, reducing the time that a message must wait to be processed. Figure 9 on page 26 is an overview of expedited message handling.

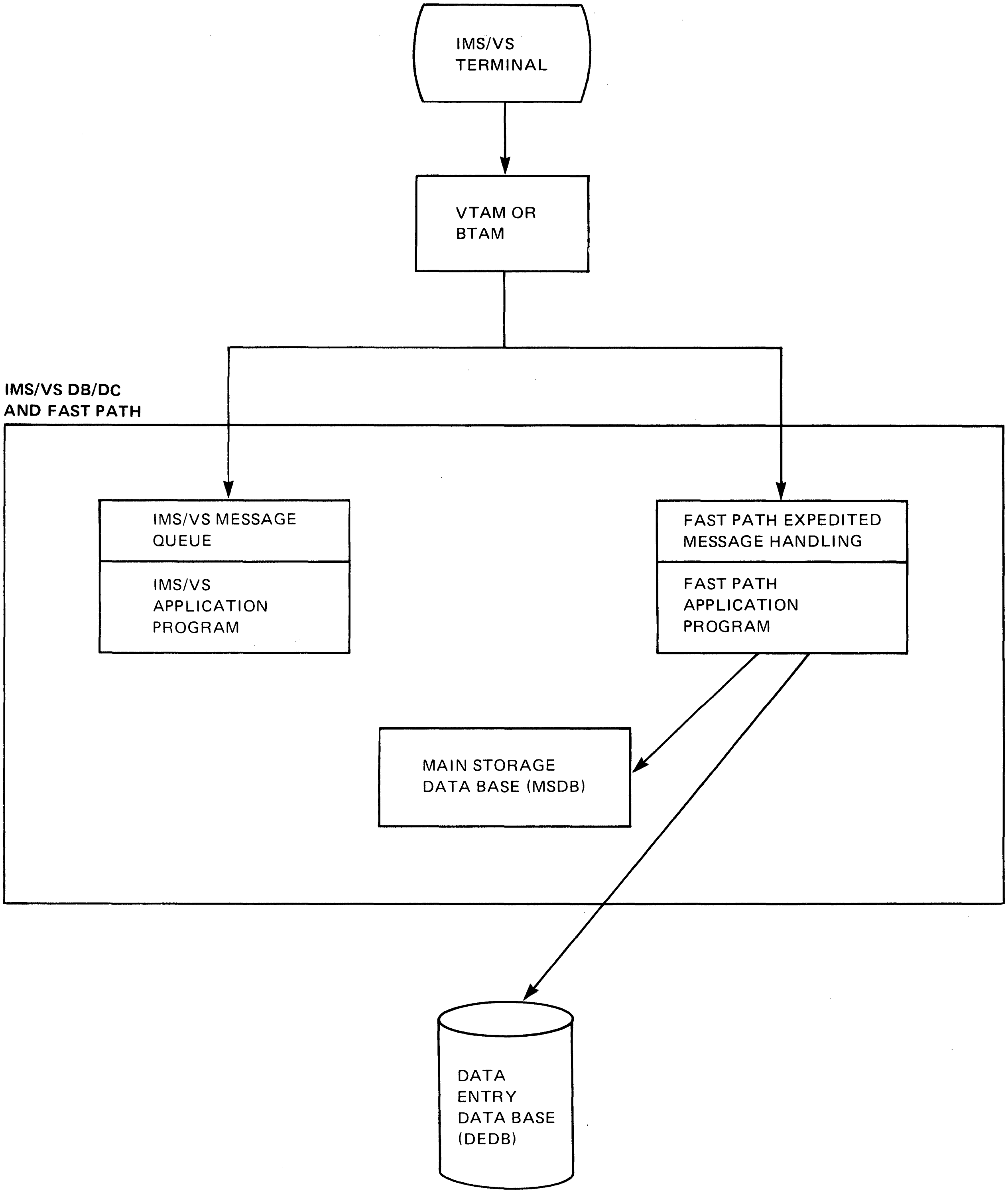


Figure 9. Expedited Message Handling

Scheduling Messages

As a part of IMS/VS system definition, you associate application programs with the transactions they will process. Thus, when a transaction is entered, IMS/VS automatically schedules the associated application program to process it.

Transactions can be assigned a message class and a priority. Based on these and other scheduling parameters, IMS/VS can automatically balance its workload by scheduling the same application program or the same transaction type into more than one IMS/VS region.

Input message queues can be assigned a limit. Once the number of messages on a given input queue reaches this predefined limit, the priority of messages on this queue is raised to improve the processing time of these messages.

By applying the parameters you have selected during system definition, IMS/VS responds to the workload by manipulating the order in which messages are processed.

Controlling Online IMS/VS Operation

The IMS/VS system is controlled by commands that can be issued by terminal users, by the master terminal operator (MTO), or by an IMS/VS facility called the automated operator interface (AOI). The set of commands available to terminal users is a subset of the commands available to the MTO and AOI. This subset permits terminal operators to affect the operation of their terminals and workload, but not the operation of the system as a whole. The sections below further describe how the IMS/VS online system can be controlled by the MTO and AOI.

Master Terminal Operator Control of IMS/VS

The master terminal is a logical terminal that acts as the operational hub of IMS/VS. The master terminal operator (MTO) has complete control of IMS/VS communication facilities, message scheduling, and data base operations. The MTO monitors scheduled operations and handles abnormal situations that require recovery actions.

From the master terminal, the MTO can enter IMS/VS commands to:

- Start and stop system resources such as data bases and terminals
- Display the status of various IMS/VS resources
- Change the parameters that control program scheduling
- Prevent further processing if an error occurs
- Schedule recovery and restart functions

IMS/VS commands permit the MTO to dynamically alter the system's operation. The MTO must therefore know all the operating aspects of the system and be familiar with the applications that share the system's resources. In addition, because the MTO controls recovery and restart processing when an abnormal termination occurs, each installation must provide a detailed set of installation-specific operating procedures for the MTO to use both during normal daily operation and during restart situations.

Automated Operator Control of Some IMS/VS Functions

IMS/VS provides support called the automated operator interface (AOI). AOI provides:

- A set of DL/I commands and associated IMS/VS processing that allow an application program to issue IMS/VS operator commands and receive responses to those commands.

- A means for copies of messages to be passed to a user-written exit routine, which routes them to any other destination, such as a logical terminal.

- Hardcopy logging of the results of a master terminal operator command. The specified output is printed at a secondary master terminal. The category of messages that will be logged can be specified as a system definition option.

The advantages of using the AOI include:

- Reduced operator intervention

- Improved accuracy of input commands, because AOI is automatic and does not rely on the operator's manual entry

- Increased information in hard-copy audit trails

The use of the automated operator interface is optional. It requires user-written programs and/or an optional user-written exit to perform specific processing your installation may require.

Additional IMS/VS Facilities

The sections below describe some facilities provided by IMS/VS that expand your system's capabilities.

Fast Path

Fast Path is a part of IMS/VS that supports applications requiring high data availability and fast processing. Although Fast Path has its own data bases and message processing, it is an integral part of IMS/VS. Fast Path is able to provide high data availability and fast processing through its:

- **Data Bases**

  Fast Path has two types of data bases, each designed to provide efficient storage for and access to a different kind of data:

  - **Main Storage Data Bases** (MSDBs) store and provide access to an installation's most frequently used data. The data in an MSDB is stored in segments. Each segment can be available to all terminals or be assigned to a specific terminal. To provide fast access and allow frequent update to this data, MSDBs reside in virtual storage during execution.

  - **Data Entry Data Bases** (DEDBs) provide a high level of availability for, and efficient access to, large volumes of detailed data.

DEDBs use a data structure that allows them to be used for both hierarchic processing and journaling.

- **Expedited Message Handling** bypasses IMS/VS message handling, allowing messages to be processed more quickly. Figure 9 on page 26 gives an overview of expedited message handling.

- **Data Integrity and Recovery Processing**

  Fast Path's message and data base processing are synchronized to protect the integrity of the messages and the data. With Fast Path processing, the data base isn't updated, nor messages sent, until the application program reaches the end of a unit of processing.

- **Utilities**

  Fast Path has utilities to create, maintain, reorganize, and recover DEDBs and MSDBs. Most of the DEDB utilities can run online, concurrently with application programs, thus helping DEDBs provide high data availability. The DEDB online utilities:

  – Extract and delete journaling or auditing information from DEDBs.

  – Reorganize DEDBs.

  – Create and compare copies of an area.

  The DEDB offline utilities:

  – Initialize and format DEDB areas.

  – Recover DEDBs.

  The MSDB utilities, which run offline:

  – Initialize and change MSDBs

  – Recover MSDBs

- **Application Programs**

  The following application programs can access MSDBs and DEDBs:

  – Message-driven programs

  – BMPs

  – MPPs

  These programs are described in more detail in Chapter 4, "IMS/VS Application Programming" on page 33.

- **Data Base Calls**

  Two calls are provided for processing MSDBs and DEDBs:

  - **The POSITION call** is useful in gathering or deleting journaling information, and in determining how much unused space exists in a DEDB area.

  - **The FIELD call** (for MSDBs only) makes it possible to access the contents of a field within a segment, then change the value in that field based on its contents.

Message Format Service

Message Format Service (MFS) is a message editor—a tool that formats IMS/VS messages. It permits application programs to deal with logical messages instead of device-dependent data. An application program that uses MFS can interact with different device types with a single set of message editing logic.

A program using MFS for device interaction does not have to be concerned with physical characteristics of the device, unless it wants to use certain, specific device features. Even when these features are used, the application program requests functions using MFS logic; it does not send or receive device control characters directly.

MFS also provides logical and physical paging for IBM 3270 and 3604 Display Stations. This permits the application program to produce large amounts of data that MFS formats for multiple screen displays at the terminal. The operator can move backward and forward between these screen displays or "pages" at the terminal.

Additionally, MFS's Distributed Presentation Management allows message formatting to be shared between IMS/VS and programmable controllers, such as IBM's Finance Communication System or Intersystem Communication subsystems.

MFS supports MVS/Extended Architecture (XA) and takes advantage of the increased virtual and real storage provided by System/370 Extended Architecture series processors.

Multiple Systems Coupling and Intersystem Communication

Facilities provided by Multiple Systems Coupling (MSC) and Intersystem Communication (ISC) enable a transaction or data to be processed in a subsystem that is not the originating one. The facilities make it possible to:

- Have access to an IMS/VS subsystem from another subsystem

- Route transactions and messages

- Distribute transaction processing

- Grow beyond the capacity of one IMS/VS system

- Respond to constraints of capacity or response time

Multiple Systems Coupling

MSC permits you to:

- Enter transactions or messages in one IMS/VS subsystem for processing in another IMS/VS subsystem. IMS/VS returns the results to the initiating terminal without the terminal operator being aware of this processing. To the operator, processing seems to have occurred in a single system.

- Access an independently operating IMS/VS subsystem by means of a terminal or program to process data or obtain a response.

- Configure multiple IMS/VS systems so that processing loads and system data bases are distributed among them to satisfy particular geographic and business requirements.

Individual system definitions describe the routing for all messages. Message routing is automatic, unless you use a routing exit routine to alter a message's destination.

MSC directed routing allows different IMS/VS systems in the same MSC network to use the same logical names for terminals and transaction codes. The application program can determine the system identification of the system that scheduled it. It can directly specify the system identification of the system that will process its message and also specify a destination within that system.

Intersystem Communication

As with MSC, Intersystem Communication permits you to communicate with and exchange data between multiple IMS/VS systems. In addition, ISC permits communications between IMS/VS and CICS/VS or a user-written subsystem, provided all the subsystems involved have implemented Intersystem Communication protocols.

IMS/VS may be either the back-end or front-end subsystem in an intersystem communication network.

As a back-end subsystem, IMS/VS application programs can be initiated from terminals or by application programs residing in another system. Similarly, when acting as the front-end subsystem, IMS/VS transactions can cause messages to be sent initiating processing in the other subsystem, with results returned to IMS/VS.

Comparing MSC and ISC

There are some similarities between MSC and ISC, but there are some important differences as well. These can be seen in the following:

MSC:

- Connects multiple IMS/VS systems to each other. The IMS/VS systems may reside in the same processor or in different processors.

- Provides processing that is not apparent to the terminal operator; processing appears to occur in a single system.

- Does not require the terminal operator or the application program to know routing information unless directed routing is being used. Routing is automatic, based on parameters supplied during system definition.

- Supports conversational transactions

In comparison, ISC:

- Can connect like or unlike subsystems, provided each subsystem to be connected implements ISC. You may couple IMS/VS to:

  - Another IMS/VS subsystem

  - CICS/VS

  - A user-written subsystem

- Allows you to use the networking facility of System Network Architecture (SNA).

- Requires involvement of the terminal operator or application program to determine the destination of a message. Specified routing parameters may be overridden or modified by Message Format Service.

- Provides a unique message switching capability that permits message routing to occur without the involvement of an application.

- Permits the use of Message Format Service to assist in routing and formatting messages between subsystems.

Chapter 4. IMS/VS Application Programming

IMS/VS application programming support provides several important advantages. Application programmers can write IMS/VS application programs in COBOL, PL/I, or assembler language—programming languages that are in common use. Although application programmers do need to learn special DL/I calls to access the IMS/VS data base and data communication calls to send and retrieve messages, these calls comprise only a portion of the application support. Because the format and syntax of the calls are the same for both message and data base processing, application programmers only need to become familiar with one interface.

IMS/VS application programs are independent of the physical storage of the data they process. Application programs are not concerned with where or how the data is stored; instead, each program uses its own view of the data, a view that contains only the data it needs, in the structure in which it needs it.

As a result, IMS/VS application programming offers productivity and flexibility to those who develop IMS/VS applications. This chapter describes some of the highlights of IMS/VS application programming.

Online and Batch Processing Environments

IMS/VS supports two application processing environments: online and batch. Online processing allows many people to use the system at the same time through terminals. Application processing in an online environment makes it possible for:

- Many users to access the information in the data base
- The application to communicate with people at terminals and other applications
- Users to invoke the application from terminals
- Users to have the results of the application's processing immediately when the application's response time is crucial
- The information in the data base to be current

Application processing in a batch environment is the kind of processing that usually isn't done during prime shift. Batch processing is typically used when you need to update a large portion of the data base, or to print a detailed report.

IMS/VS Application Programs

IMS/VS supports three kinds of application programs in the online environment: MPPs, message-driven programs, and BMPs. IMS/VS also supports programs that run exclusively in the batch environment. Online application programs can perform message processing or batch-type processing; programs in the batch environment can perform only batch-type processing.

Message Processing Programs: MPPs

Message processing programs, or MPPs, are IMS/VS application programs that process messages. When a person at a terminal enters a request for information, the request is called a message. IMS/VS schedules a program when there is a message for it to process. The program then retrieves its messages one at a time, processes them, and replies to the terminals that sent the messages. MPPs:

- Are designed to respond to processing; their main purpose is to provide answers to requests.

- Can merely answer requests for information from the data base or make changes to the data base.

For example, suppose a bank teller enters a request (a message) at a terminal for the current balance in a certain checking account. Part of the message is a transaction code that identifies the processing request and associates the request with the application program that can answer the request. The program retrieves the message, gets the requested information, and replies to the bank teller's terminal.

MPPs can access all IMS/VS data bases, including DEDBs and MSDBs. Unlike BMPs and batch programs, MPPs cannot access OS/VS files and GSAM data bases.

Message-Driven Programs

As with MPPs, message-driven programs are IMS/VS application programs that process messages. The main differences between message-driven programs and MPPs are:

- Messages processed by message-driven programs must consist of only one segment. Messages processed by MPPs can consist of several segments.

- Message-driven programs use Fast Path's expedited message handling, and bypass IMS/VS scheduling, allowing for efficient processing.

Message-driven programs can access all IMS/VS data bases, including DEDBs and MSDBs.

Batch Message Processing Programs: BMPs

Online programs can also perform batch-type processing. IMS/VS application programs that perform batch processing online are **batch message processing programs**. BMPs have characteristics of programs in both online and batch environments in that they run online, but, as with programs in a batch environment, are started with job control language (JCL). BMPs do not have to receive their input from terminals; their input can be from an OS/VS file. BMPs don't necessarily process messages (although they can); BMPs are typically used to perform batch-type processing online.

BMPs can access the data base concurrently with MPPs, even if your installation doesn't use data sharing. (If an installation uses IMS/VS data sharing, it's possible for MPPs, BMPs, and batch programs to access the same data base concurrently.) If you don't use data sharing, a DL/I data base may be accessed by programs running in an online environment or in a batch environment, but not both concurrently. The use of BMPs can circumvent this problem, and and allow you to share data with MPPs.

Although BMPs are generally used to perform batch-type processing online, they can send or receive messages. There are two kinds of BMPs:

- A **transaction-oriented BMP** accesses message queues for its input and output. It can also process input from OS/VS files, and it can create OS/VS files.

- A **batch-oriented BMP** does not access the message queue for input; it is merely a batch program that runs online. It can send its output to an OS/VS output device.

In addition to accessing all IMS/VS data bases, a BMP can access GSAM data bases and OS/VS files.

Batch Programs

Programs that run in a batch environment do not process messages. Because a batch program runs by itself—it doesn't compete with any other programs for IMS/VS resources. It's possible to run a DB/DC system and a batch system at the same time; they just can't access the same data bases (unless you use data sharing). Because batch programs are typically longer-running programs than online programs, you usually don't run them during prime shift; instead you run them when the online system isn't being used (for example, on the weekends), so that you don't tie up system resources when there are a lot of programs or users needing the computer's resources. Batch programs typically produce a large amount of output, and are not invoked by another program or user. They are usually scheduled at specific time intervals, for example, weekly, and are started by JCL.

You can write batch programs in such a way that you can run them as batch programs or as BMPs; running the program at different times does not affect its structure or code.

A batch program can access most IMS/VS data bases, OS/VS files, and GSAM data bases. Unlike MPPs, BMPs, and message-driven programs, a batch program cannot access DEDBs or MSDBs.

Processing the Data Base: Data Language/I Calls

IMS/VS application programs read and update DL/I data bases by issuing calls to DL/I. By using DL/I calls, application programs can retrieve, replace, delete, and add segments to the data base.

A DL/I call is made up of a call statement and a list of parameters. The parameters for a call give DL/I information that it needs to execute the call&emdash.for example, the call function (or what you want done), the name of the data structure that DL/I should use for the call, the data area in the program into which DL/I should return the data, and, if you wish, a condition that the data to be retrieved must meet.

Processing Messages: Data Communications Calls

MPPs and BMPS retrieve and send messages by issuing calls similar to DL/I calls. These calls are called data communication calls, or DC calls. By using DC calls, application programs can retrieve input messages, send output messages to the terminal that sent the input message, and send output messages to other application programs and terminals.

Authorized application programs can also issue IMS/VS commands and receive responses to those commands using two additional DC calls. This helps to reduce operator intervention, and is explained in "Automated Operator Control of Some IMS/VS Functions" on page 28.

Editing IMS/VS Messages

IMS/VS provides several message editors to assist in processing messages. These are IMS/VS basic edit, Message Format Service, and Intersystem Communication (ISC) edit. In addition, you may write your own edit routine, should it be required.

Basic edit helps to simplify application programming by removing the control characters from an input message before the application program receives it and by inserting the control characters you specify when the application program sends a message back to the terminal. Basic edit is available to all IMS/VS application programs.

Message Format Service is a message editor that assists in making application programs independent of the devices they will use for input and output.

When an application program communicates with a terminal, Message Format Service (MFS) translates messages from the way they are entered at the terminal to the way the program expects to receive and process them. By formatting messages between the terminal screen and the application program, MFS reduces the work that the program must perform, and simplifies application program logic.

MFS uses control blocks that define what a message should look like to the person at the terminal and to the application program. These control blocks contain information about how the data should look when it's passed between the application program and the terminal.

ISC edit is a special editor used for Intersystem Communication sessions between IMS/VS and other subsystems linked in an ISC network. It is used to edit transactions, commands, and message switches.

Application Development Tools

There are some tools available to IMS/VS users to help them in developing and testing their applications. This section describes some of these aids.

The IBM DB/DC Data Dictionary

The Data Dictionary (Program Product 5740-XXF) is a development and administrative tool you can use to manage information about an installation's data processing and other data resources. You can use the Dictionary to document the following application data:

- Data elements
- Data bases
- Programs
- Transactions

Information in the dictionary can be used to analyze the impact of proposed changes, and to control the changes as they occur.

You can also use the Dictionary to define and generate DBDs and PSBs, and application program data structures; this allows you to enforce consistent programming standards and naming conventions.

The IMS/VS Applications Generator: Application Development Facility

The IMS/VS Application Development Facility (ADF; Program Number 5796-PHX) is designed to help IMS/VS users reduce the time, cost, and risk in developing and maintaining IMS/VS applications. Running as an application program under IMS/VS, ADF interprets specifications and executes applications, making it possible for many IMS/VS applications to be developed and put into production without conventional programming.

The advantages of using ADF include:

- ADF can reduce the work required to develop and maintain application systems.

- ADF can help your organization respond to change.

ADF works by providing an application structure, data bases for storing specifications, and several processing function modules, for example, transaction drivers and utilities.

Testing an Application: Batch Terminal Simulator II

Batch Terminal Simulator II (BTS II; Program Number 5796-PGT) is useful in testing all types of IMS/VS application programs. One of the advantages of using BTS is that it allows you to test online application programs without running them online. BTS II can also test call sequences, and it provides information about each transaction as the transaction goes through the IMS/VS system. This information is helpful in debugging.

Testing DL/I Call Sequences: The DL/I Test Program

The DL/I test program is an IMS/VS application program that executes the DL/I calls you specify against a test data base. An advantage of using the DL/I test program is that you can test the DL/I calls in your program without executing the program as a whole. For each call, the DL/I test program checks both the access path you've defined for the segment and the effect of the call. The DL/I test program is helpful in debugging, because it can display DL/I control blocks. You can also use the DL/I test program to time call sequences to check on their performance.

Highlights of IMS/VS Application Programming

This chapter has explained some of the reasons that IMS/VS can offer productivity and efficiency to application programmers. IMS/VS handles many aspects of application programming that are usually left to the programmer. For example:

- IMS/VS application programs can be independent of the way in which data is stored. Data and the way in which the application program accesses it are defined outside the application program itself.

- IMS/VS provides a means to simplify application programming by providing a call structure that is common in format and syntax for both data base access and message processing.

- IMS/VS provides editors to make online programs independent of terminal and line protocols. IMS/VS-provided editors also relieve the application programmer of the need to edit messages before presentation to either the application program or to the output device.

- IMS/VS provides calls and utilities that assist in maintaining data integrity.

- IMS/VS provides aids that help programmers in developing, testing, and debugging their application programs.

Chapter 5. System Control Facilities

IMS/VS provides many facilities that aid you in defining and controlling your IMS/VS system. These facilities, which we will call system control facilities, support both the Data Base and Data Communications functions. This chapter discusses the way in which IMS/VS provides for:

- Defining your IMS/VS system
- Modifying your IMS/VS system
- Securing your IMS/VS system
- Recovering from failures
- Monitoring and tuning your IMS/VS system
- Defining IMS/VS as a portable system

Defining your IMS/VS System

IMS/VS is defined using a set of macro instructions. Whether you have a batch-only system or an online system, you use these macro instructions to define your IMS/VS system environment and data bases. If you have an online system, you also use these macros to define applications, logical and physical devices, multiple system configurations, and message routing information.

IMS/VS system definition is a two-stage process, with an optional preprocessor. Stage 1 of system definition checks your input specifications and generates a series of OS/VS job steps that are input to Stage 2. Stage 2 builds IMS/VS system libraries, execution procedures, and the IMS/VS online control program tailored to support your desired set of IMS/VS functions and stores these in an IMS/VS library. The optional preprocessor, run before Stage 1, permits you to define a larger IMS/VS system without a proportional increase in system generation time. The preprocessor accomplishes this by performing certain checks before actual assembly of your system definition Stage 1. These include a check for duplicate resource names among the names you have defined and a check to ensure that the resource names you have defined are of the correct length and format.

Although the initial installation of an IMS/VS system establishes libraries and OS/VS operating system options, the system definition portion is subject to change as a result of your requirements to change your network configuration, to enlarge the system, or to tune it. Thus, the Stage 1 system definition input becomes a master control point for system definition maintenance. The existence of this Stage 1 input ensures a consistent and available set of documentation of your IMS/VS system's structure and function.

A special system definition capability permits you to generate different versions of the IMS/VS online system by means of a suffix appended to control modules in the IMS/VS nucleus and certain control blocks. This capability enhances the flexibility of IMS/VS by:

- Permitting the creation of a test system having characteristics that are the same as, or similar to, your production IMS/VS system. Thus, new applications or transactions may be tested and corrected in an actual production environment without impacting your production system.

- Permitting the inclusion of certain IMS/VS facilities, such as Fast Path, in one version of the system without the necessity of including these features in other versions.

Modifying Your IMS/VS System

You do not have to perform a complete system definition each time your system changes. Many changes, particularly those required to tune the system, may be made using job control language (JCL) parameters at system run-time. This capability is described in the section below entitled "Modifying Execution-Time Parameters." However, the addition of application programs, data bases, and transactions, as well as physical changes to your IMS/VS network, do require redefinition of some or all of your IMS/VS system. Adding, changing, or replacing application programs, data bases, and transactions, and MFS formats as well, may be done without stopping your IMS/VS system to redefine them. These changes may be made online. This capability is discussed further in the subsequent section "Changing System Resources Online" on page 41.

Modifying Execution-Time Parameters

After you define your base IMS/VS system and build the basic IMS/VS libraries, your system is ready to use. Several sets of control parameters, stored as members of the IMS/VS procedure library, control your execution. To modify your IMS/VS system, you can either rerun your system definition or directly modify members of the IMS/VS procedure library.

Although many of the procedures generated and placed in the IMS/VS procedure library require alteration before they can be used in a direct execution of IMS/VS, they provide a convenient starting point for the task of defining execution JCL. The content of many of the members directly reflects the options that you specified during your Stage 1 system definition input.

After you have stored the members in the procedure library, you can name them to fit your installation's requirements. The names could, for example, follow a convention that indicates ownership by a particular application or that relates to a sequence of execution. The procedure names are used by either the system operator or the IMS/VS master terminal operator to invoke execution of the associated OS/VS job.

Then, you can tailor the procedures to your installation's requirements. There are a number of initialization parameters that can be specified for your startup procedure. You specify them either as parameters on the IMS/VS execution (EXEC) statement or as parameters in a member of the procedure library.

The ability to modify and control your IMS/VS system at execution-time allows you to adjust to changing application and system requirements, to conveniently monitor and tune system performance, and to respond to user requests in a timely fashion. All of this improves your operational flexibility.

Changing System Resources Online

In many installations, it is important that the online system be available to users during a large portion of the day. Users who require their online IMS/VS system to be available continuously need to be able to modify that system online. The ability to add, delete, and replace the following IMS/VS resources without bringing down your IMS/VS system is a major step toward continuous operations:

- Data base definitions and directories
- Application definitions and descriptors
- Transaction definitions
- MFS formats
- Security definitions
- Fast Path routing code definitions

In general, changing your system online involves the following steps:

1. Perform a subset of the control blocks system definition.
2. Run the Online Change utility.
3. Perform a sequence of operator commands.

IMS/VS automatically monitors the resources that are currently active in your system, and allows you to return to the unchanged, back-level system should an error occur.

After an online change is successfully completed, it persists across all types of IMS/VS restarts.

Securing Your IMS/VS System

Although it is important that users have ready access to the IMS/VS resources they require, it is equally important that IMS/VS resources be restricted from unauthorized use. IMS/VS security facilities provide the means to balance these needs.

IMS/VS system definition creates basic system security, which prohibits the entry of certain commands from any terminal other than the IMS/VS master terminal.

IMS/VS also offers you extended security. Extended security includes the following levels of security:

- Terminal security—Defines the commands and transactions that may be used from a given physical or logical terminal.
- Password security—Limits the use of a specified IMS/VS resource to someone who supplies the correct password.
- Sign-on verification security—Defines to IMS/VS a list of terminals that require sign-on verification.

- Resource access security—Prevents an IMS/VS resource from being used by an application program unless that resource has been authorized for use by that application program.

- Transaction command security—Permits an application program to issue a subset of IMS/VS operator commands.

Depending on your operating system and the kind of extended security you wish to implement, you will use one or more of the following tools:

- The IMS/VS SECURITY macro statement, which is a part of the IMS/VS system definition

- The IMS/VS Security Maintenance Utility

- The Resource Access Control Facility (RACF) Program Product (available on an OS/VS2 MVS system only)

- User exit routines

What Resources Can Be Protected?

Using the tools enumerated above, you can protect the following IMS/VS resources:

- IMS/VS Data Communication itself can be protected from unauthorized use.

- The IMS/VS control region and IMS/VS dependent regions (MPPs, BMPs, and IFPs) can also be protected.

- System data sets may be protected by OS passwords or, if VSAM data sets, by VSAM data set protection.

- IMS/VS data bases may be secured by defining passwords for a given data base, and by implementing segment level or field level sensitivity.

  IMS/VS data bases may also be secured by careful definition of the program specification blocks (PSBs) of the application programs that will access the data. Appropriate PSB definition can prohibit an application program from accessing any data, except that which it needs to complete its processing, and can control authorization to update or delete data.

- Physical terminals can be protected by requiring that sign-ons to a given terminal be verified to ensure that the person signing on is authorized to use that terminal. Additionally, user IDs can be associated with passwords, to add password protection for the terminal.

- Logical terminals (LTERMs) can be protected by limiting a terminal's access to IMS/VS resources to an eligible set of transactions or commands. Additionally, passwords may be associated with each transaction or command.

- Transactions and commands can be protected by identifying the set of LTERMs that are authorized to enter them, and by associating a password with each transaction or command.

- PSBs can be protected from use by an unauthorized dependent message region.

- Online application programs can be protected by passwords. Like PSBs, application programs can be protected from use by a dependent message region unless that region has been specifically authorized to use them.

Recovering from Failures

IMS/VS has full recovery facilities to protect your installation should one of the following fail:

- The IMS/VS system
- An application program running under IMS/VS
- An IMS/VS data base
- The operating system, network control program, telecommunications control program, etc.

IMS/VS is a DB/DC system designed to be recoverable. Much of the processing required to recover from failures of the first three types is automatically performed by IMS/VS. This processing requires little, if any, user intervention. While failures of components described in the last bullet above are outside of the normal recovery processing of IMS/VS, IMS/VS recovery mechanisms shield the IMS/VS system from these failures to a great degree. IMS/VS provides several mechanisms that assist you to recover in the event of a failure—that is, to reconstruct lost information or to remove errors you have detected from existing information. Some of these mechanisms are automatic and some require human intervention. These mechanisms include:

- **Checkpoints**

  IMS/VS automatically creates periodic checkpoints by recording current status at the system level. The master terminal operator may also cause IMS/VS to take a system checkpoint. This gives the MTO additional control over the currency of the information that may be needed for recovery and may shorten the time needed to recover from a failure.

  Applications can be designed to take checkpoints that are local to that program. A checkpoint in an application program serves a dual purpose. It marks an intermediate completion point in the application program's processing—a place where the work finished so far may be judged correct. If recovery is required before the next checkpoint is taken, the recovery process need go no further back than this point. An application program checkpoint also serves to release any resources held for the updating program and causes any output messages the program has created to be enqueued for output.

  Backup copies of data bases can be considered a form of checkpoint, as can backup copies of input and output message queues. IMS/VS offers several utility programs to assist you in making backup copies of your IMS/VS data bases. You can also make periodic backup copies of your system data sets by using utilities provided by your operating system.

  IMS/VS can be instructed to make backup copies of its message queues when it shuts down. However, if you run IMS/VS continuously for extended periods, you may want to take online copies of your message queues periodically to shorten the time necessary to recover should a message queue be lost. By means of a master terminal operator command, you can cause IMS/VS to make a copy of a message queue without shutting down.

- **Message queues**

  Messages that are input to IMS/VS are written to input message queues before being processed by an IMS/VS application program. Output messages are also written to queues before being transmitted to their final destination, such as a user terminal. When defining a transaction during system definition, you can specify whether that transaction will always be recoverable. (With expedited message handling, input messages are always nonrecoverable; output messages are recoverable.) If a transaction is specified as recoverable, any message associated with that transaction is not removed from a queue until IMS/VS receives an acknowledgment that the message has reached its destination. By this requirement, IMS/VS ensures that these messages can be recovered across system failures and telecommunication session failures. IMS/VS can determine the message's status and cause it to be reprocessed or re-sent to its destination without user intervention.

- **Emergency restart**

  Emergency restart is an extension of IMS/VS's normal restart process. It is initiated by a master terminal operator command whenever it is necessary to restart IMS/VS after an IMS/VS, OS/VS, hardware, or power failure, or whenever a prior execution of the IMS/VS system was not terminated with a successful checkpoint.

  Emergency restart involves resetting the data bases to their condition at the time of failure, restoring message queues to their condition at the time of failure, and resetting each active region to its last **synchronization point**—that is, the point at which processing in that region was known to be correct and complete.

- **Logs**

  Logs are records of activity—lists of work done and changes made to the system or its resources. Batch IMS/VS users can record this activity on their system log. This log may be on direct access storage devices (DASD) or tape and is a record of the activity of the batch system. Online users use direct access storage devices to hold the record of their online activity. These DASD data sets may later be archived to a disk or tape system log.

  In contrast to the taking of certain checkpoints, which involves human activity, the writing of logs is handled totally by IMS/VS. Data Base Recovery Control (DBRC) will keep track of the status of the online log and system log data sets. If you request it, IMS/VS can automatically create the JCL necessary to run a utility that will archive the online logs to the system log(s) at intervals you can specify. When restart is required, DBRC automatically determines and makes available the correct data sets needed for recovery.

- **Automatic backout**

  If a failure occurs during the running of a program, the updates made to the data base by that program may be only partially complete. Backout of partially-complete (uncommitted) changes made by the IMS/VS online system is handled automatically by IMS/VS and does not usually require human intervention. (Backout is not performed if you're using Fast Path, because the data base is not physically changed until commit processing.)

If log records are stored on direct access storage, you can also use automatic backout for a batch system.

- **Utilities and other aids**

  IMS/VS provides a comprehensive set of utility programs that assist you in recovering your IMS/VS system in the event of a failure. These include:

  - The **Log Archive utility**

    A utility that archives the log(s) created by your IMS/VS system

  - The **Log Recovery utility**

    A utility used to recover a log data set that has encountered an I/O error or has not been successfully closed.

  - The **Data Base Backout utility**

    A utility that uses the batch system log (tape or disk) to back out changes made by a batch region. It is used if automatic backout is not available.

  - The **Data Base Change Accumulation utility**

    A utility that accumulates data base records only into a sequential data set. These records can be used to recover a data base and may shorten the time required to recover.

  - The **Data Base Recovery utility**

    A utility designed to restore a physically damaged data set within an IMS/VS data base by updating a copy of the data base with changes logged after the copy was made.

  - **Log data formatting utilities**

    Various utilities that format and analyze the data contained on the system log. The log analysis utilities may be used to create information that you may use to monitor and tune your IMS/VS system. This is discussed later under "Monitoring and Tuning Your IMS/VS System."

Thus, IMS/VS's support for restart and recovery offers you the following advantages:

- Much of the tracking and checkpointing required to successfully restart the system and recover after a failure is automatic. You must still plan carefully for recovery, but the tools to do the job are available to you and you can control their use. That is, by means of system parameters, you can balance the system overhead of your recovery mechanisms with your processing and performance requirements.

- Logging to DASD provides for simpler, more automatic restart procedures and eliminates the tape handling formerly required for the online IMS/VS system.

- End users and operators of IMS/VS are shielded to a great degree from the need to activate recovery processing.

Monitoring and Tuning Your IMS/VS System

After your IMS/VS system is up and running, you will want to observe its performance and then adjust it to achieve the performance and response time criteria you have established. IMS/VS provides a number of tools to support these activities. The system log is a basic source of statistics about the actual processing of the online system. Individual log record types contain data that may be analyzed in many ways. For example, all activity related to a specific user ID can be selected and printed, as can statistics about IMS/VS pools.

IMS/VS provides several utilities to assist in extracting log record types from the system log, and reducing and merging data spread across several log volumes. These utilities include:

- The File Select and Formatting Print program

  Prints or copies the contents of log records contained in the input system log volumes.

- The Log Transaction Analysis utility

  Produces a report of information about individual transactions.

- The Statistical Analysis utility

  Produces several summary reports about actual transaction loads and response times.

The IMS/VS DB Monitor provides a trace of internal activities of an IMS/VS DB system and a program that produces summary and detailed reports from this collected data. The DB Monitor assists in analyzing application designs, data base designs, and resource allocation.

The IMS/VS DC Monitor, which is part of the online control program, collects data and produces a series of reports that allow you to keep track of:

- Scheduling and termination events

- Activity between application programs and message queues

- Activity between application programs and data bases

A wide range of system-level events can be recorded and reported on by the Generalized Trace Facility (GTF), which is available with the OS/VS operating system. Additionally, the Program Isolation Trace Report, produced by an IMS/VS utility, can be used to measure contention for a given data base segment.

After you have put procedures in place to monitor your IMS/VS system and have obtained performance statistics, you can begin to tune your system. Performance management, which includes monitoring and tuning, is an ongoing process, as illustrated in Figure 10.

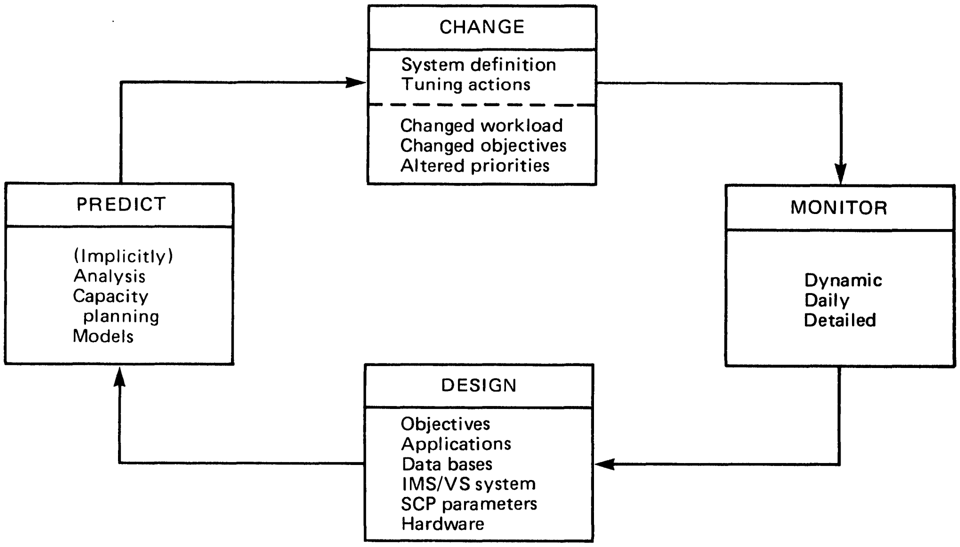


Figure 10. Activity Cycle for IMS/VS Performance Management

IMS/VS provides a number of ways in which system performance can be tuned to meet your specified criteria. Among these are:

- Optimization of IMS/VS buffer pools
- Optimization of application module loading
- Minimizing contention for IMS/VS resources

Defining IMS/VS as a Portable System under OS/VS2 MVS

IMS/VS can be defined as a **portable** or a **nonportable** system. A portable system is one that will run on either a System/370 or a System/370 in Extended Architecture (XA) mode. A nonportable system will run **only** on a System/370 in XA mode. (To take advantage of the MFS support for XA, your system must be defined as nonportable.)

System/370 XA requires OS/VS2 MVS/XA to be installed. This comprises OS/VS2 MVS/System Product Version 2 Release 1 and Release 1 of the Data Facility Product (DFP).

Defining IMS/VS as a portable system can ease the workload of migrating an IMS/VS system to the XA environment.

Chapter 6. Using IMS/VS with Other Subsystems

IMS/VS subsystems may be linked into a network using Multiple Systems Coupling. By using Intersystem Communication, IMS/VS can become part of a network that contains CICS/VS subsystems or user-written subsystems. These topics are discussed earlier in Chapter 3, "IMS/VS Data Communication" on page 23. This chapter discusses other ways in which IMS/VS can be coupled to non-IMS/VS subsystems to extend your processing power and productivity.

Using DL/I with CICS/VS

If your network contains CICS/VS, you can use that system to manage your terminal network while accessing IMS/VS data bases. You can use the following application programs to access data in an IMS/VS data base:

- CICS/VS online programs. An online program accesses resources concurrently with other online resources, and is useful when response time is important, the information in the data base must be available to many users, and the program needs to communicate with terminals.
- IMS/VS batch programs. A batch program is useful for producing large amounts of output at specific intervals, when response time is not critical.
- CICS/VS shared data base programs. A shared data base program is an IMS/VS batch program that performs batch-type processing of an online data base. Although it doesn't share data directly with online programs, it can process a data base concurrently with CICS/VS online transactions. If your CICS/VS release supports data sharing, you don't need to use the shared data base facility; your CICS/VS online programs can share data directly with IMS/VS batch or online programs. Figure 11 is an overview of a possible data sharing configuration.

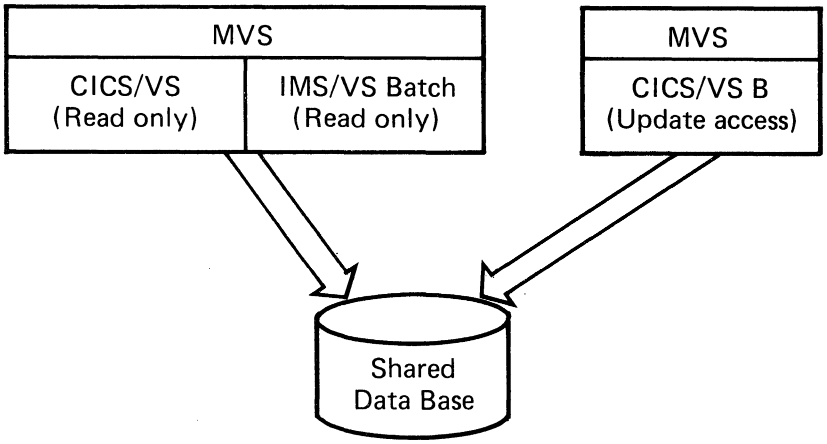


Figure 11. IMS/VS and CICS/VS: Overview of Data Sharing at the Data Base Level

CICS/VS command-level programming is used to access CICS/VS resources such as terminals, temporary storage, and transient data. You can use either DL/I calls or command-level programming (EXEC DLI) to access IMS/VS data bases. EXEC DLI is similar in syntax to the CICS/VS command language, and is straightforward and easy to learn.

Using IMS/VS with Database 2 (DB2)

The IMS/VS online system consists of hierarchic data bases managed by DL/I; terminal management is provided by the IMS/VS Data Communications Feature. You can also use the IMS/VS Data Communications Feature as the terminal manager when accessing IMS/VS Data Base 2 (DB2) relational data bases.

Transaction-driven applications can access both DB2 and IMS/VS data bases.

Figure 12 is an overview of how IMS/VS and DB2 are related.

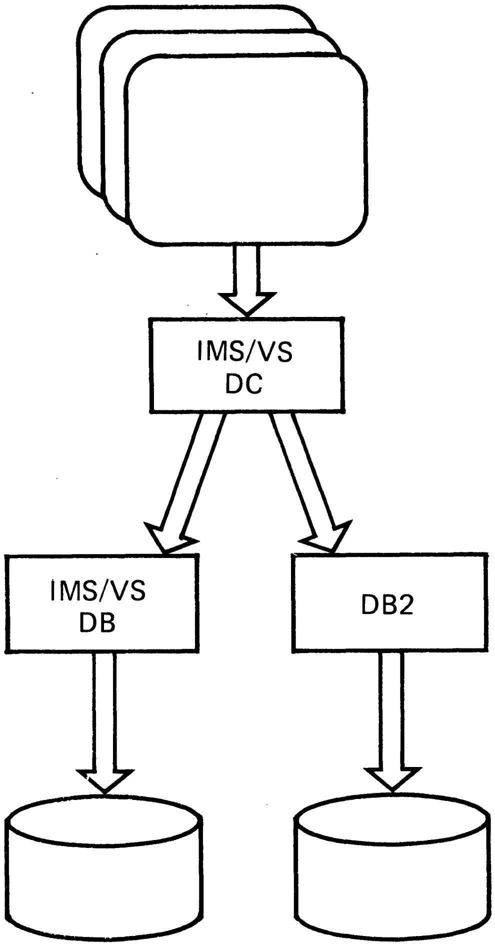


Figure 12. IMS/VS and DB2

In contrast to a hierarchic data base structure, such as DL/I, a relational data base structure is one that users perceive as a collection of tables. (A sample employee table is shown in Figure 13.)

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	SALARY
000090	EILEEN	JONES	E11	28750
005000	JOHN	SMITH	J20	18850
128746	HARRY	MANNERS	B14	24220

Figure 13. The Employee Table

DB2 uses the Structured Query Language (SQL) as its data manipulation language. You can include SQL statements in your IMS/VS message processing programs (MPPs), batch message programs (BMPs), and Fast Path programs in the same way as you include DL/I calls. IMS/VS batch programs cannot access DB2 data bases, however.

You may want to use DB2 data bases for applications that:

- Must support unanticipated rather than predefined queries
- Require dynamic data definition
- Are migrating from old sequential files into the data base environment

Chapter 7. Tasks in Using IMS/VS

This chapter describes the tasks involved in using IMS/VS, from installing the product to servicing it. One major task is identified in each section, with information on what the task involves, subtasks the major task includes, and, where appropriate, IMS/VS tools that can simplify the task or subtasks.

This chapter contains these sections:

- Planning for IMS/VS before installation
- Installing IMS/VS
- Designing and coding IMS/VS application programs
- Administering IMS/VS data bases
- Administering the IMS/VS system
- Developing IMS/VS operating procedures
- Operating the IMS/VS system
- Servicing IMS/VS

Planning for IMS/VS before Installation

Planning for installation requires a thorough understanding of IMS/VS and the operating environment where IMS/VS is to be installed. Such planning is a general task that entails making any necessary preparations for the other major tasks identified in this chapter, as well as such tasks as:

- Determining requirements for new software and hardware
- Training personnel and assigning duties
- Modifying operating environment standards, job control language procedures, and establishing naming conventions to accommodate the installation
- Evaluating the applicability of tools associated with IMS/VS

Installing IMS/VS

Installing IMS/VS is the process of making it ready for use and defining the necessary resources. This is essentially the implementation of the decisions made by the data base and system administrators. Among the subtasks that installation comprises are:

- Establishing the hardware and software environment
- Building the IMS/VS distribution libraries
- Loading the initial IMS/VS data bases

- Coding system definition macros, data base macros, and the dynamic allocation macro
- Performing system generation
- Using control statements to define communications and I/O buffer pools
- Running the IMS/VS sample problem
- If installation involves migration to a new release, running the installation regression test

Designing and Coding IMS/VS Application Programs

Application programming is the task of creating programs to meet user needs. The task of application programming includes such subtasks as:

- Analyzing user requirements
- Providing information to data base administration that will assist in designing a physical data base
- Structuring (flowcharting) the processing logic for application programs
- Designing and coding the programs
- Testing and debugging the programs
- Documenting the utilization and run procedures of the programs

Administering IMS/VS Data Bases

Administering the data base involves many tasks that collectively determine how a company's data bases can best be designed, allocated, and used to meet the processing goals of the installation. Data base administration includes the subtasks of:

- Analyzing user requirements
- Designing data bases
- Developing test data bases
- Implementing data base design
- Loading data bases
- Monitoring and tuning data bases
- Modifying data bases
- Recovering data bases

- Establishing data base security
- Setting up standards and procedures

Administering the IMS/VS System

Administering an IMS/VS system includes the subtasks of:

- Analyzing user requirements
- Designing and modifying the IMS/VS system, including updating system definitions with online changes
- Monitoring and tuning the IMS/VS system
- Establishing standards and procedures for:
  - system operations
  - online security
  - accounting

Developing IMS/VS Operating Procedures

The administrator develops procedures that describe how to operate IMS/VS. One set of procedures is intended for the use of remote terminal users who work with IMS/VS; the other set is used by the master terminal operator to run the system, as described below in "Operating the IMS/VS System."

Operating the IMS/VS System

Operating IMS/VS involves putting into practice the procedures written by the system administrator. This is done by the IMS/VS master terminal operator, whose duties include:

- Starting, stopping, controlling, and monitoring the IMS/VS system and resources
- Maintaining IMS/VS with the maintenance utilities
- Correcting errors and restarting the system after failure

Servicing IMS/VS

IMS/VS program service is the task of identifying and correcting failures in IMS/VS. (This task does not include identifying and correcting hardware problems or user errors.)

Problem identification involves more than the term implies; it includes:

- Using error messages, dumps, and service publications to identify the problem by (1) type and (2) area of failure within the program logic

- Constructing a keyword string that describes the problem, in order to search the software support data base for a description of a similar problem

- Reporting the problem for correction

Problem correction is usually the responsibility of IBM Field Engineering Program Service personnel.

Chapter 8. IMS/VS Publications

This chapter tells where to find information in IMS/VS manuals. This *General Information Manual* provides an overview of IMS/VS. Details of the current release are found in the *Release Guide*. *IMS/VS Primer Function Installation Guide* provides an introduction to the new user and suggests a starter set of functions. Both the *IMS/VS Release Guide* and the *IMS/VS Primer Function Installation Guide* provide references to the rest of the IMS/VS library. The *Release Guide* points to information for new functions; the *Primer Function Installation Guide* points to information of particular interest to the new user of IMS/VS.

The publications in this chapter are grouped by the major tasks described in Chapter 6, "Using IMS/VS with Other Subsystems" on page 48.

Planning for IMS/VS before Installation

- *IMS/VS 1.3 Release Guide* (SH20-9207)
- *IMS/VS Primer Function Installation Guide* (SH20-9208)

Installing IMS/VS

- *IMS/VS Installation Guide* (SH20-9081)
- *IMS/VS Primer Function Installation Guide* (SH20-9208)

Designing and Coding IMS/VS Application Programs

- *IMS/VS Application Programming* (SH20-9026)
- *IMS/VS Application Programming for CICS/VS Users* (SH20-9210)
- *IMS/VS Application Programming Reference Summary* (SX26-3727)

Administering IMS/VS Data Bases and the IMS/VS System

- *IMS/VS Data Base Administration Guide* (SH20-9025)
- *IMS/VS System Administration Guide* (SH20-9178)
- *IMS/VS Message Format Service User's Guide* (SH20-9053)
- *IMS/VS Utilities Reference Manual* (SH20-9029)
- *IMS/VS System Programming Reference Manual* (SH20-9027)
- *IMS/VS Programming Guide for Remote SNA Systems* (SH20-9054)

Developing IMS/VS Operating Procedures and Operating the IMS/VS System

- *IMS/VS Operations and Recovery Guide* (SH20-9209)
- *IMS/VS Operator's Reference Manual* (SH20-9028)
- *IMS/VS Data Base Recovery Control: Guide and Reference* (SH35-0027)

Servicing IMS/VS

- *IMS/VS Messages and Codes Reference Manual* (SH20-9030)

  Note: The following four manuals are licensed and are restricted materials of the IBM Corporation.

- *IMS/VS Failure Analysis Structure Tables (FAST) for Dump Analysis* (LY20-8050)
- *IMS/VS Diagnostic Aids* (LY20-8063)
- *IMS/VS Program Logic Manual* (LY20-8069)
- *IMS/VS Data Base Recovery Control: Logic* (LY35-0028)

General Reference

- *IMS/VS Master Index and Glossary* (SH20-9085)

Planning for IMS/VS before Installation

IMS/VS Release Guide for Version 1 Release 3

This manual describes the changes to IMS/VS for Release 3. The intent of the *Release Guide* is to help in evaluating Release 3 of IMS/VS, and in planning for IMS/VS Release 3 before installation. The book gives particular attention to the changes that require significant planning for installation and use, and that require migration from facilities provided in IMS/VS Release 2 and earlier releases. For each Release 3 change to IMS/VS, this manual provides a general description of the change; an explanation of the new concepts; planning information; information on how the change affects administration, installation, and operations; and migration considerations. This manual is primarily for experienced users of IMS/VS.

IMS/VS Primer Function Installation Guide

This manual is described under the task "Installing IMS/VS."

Installing IMS/VS

IMS/VS Installation Guide

Made available with the Data Base system, this manual contains information required to prepare for and install an IMS/VS system. It is organized around the steps involved in installing IMS/VS and includes:

- A step-by-step, quick reference guide for all tasks required to install IMS/VS
- Detailed information directly related to each installation step

Readers should have a good understanding of IMS/VS and of the OS/VS system generation process.

Program Directory

This is an informal document designed to provide, where pertinent, additional information for the installation process. Note that it accompanies the machine-readable material you receive, and that you cannot order it separately.

IMS/VS Primer Function Installation Guide

This manual is for new users of IMS/VS. It describes the primer sample programs. It also contains information on how to print the primer sample listings, how to obtain the sample programs from the distribution tape, and how to install the sample programs. Readers should have a basic knowledge of IMS/VS. Note, also, that this book contains information for CICS/VS users.

Designing and Coding IMS/VS Application Programs

IMS/VS Application Programming

This manual describes the tasks involved in developing IMS/VS applications. The first part of the book is a guide to application design, and discusses analyzing application data and processing requirements, and gathering requirements for data base and data communications options. The second part is a guide to application programming. It explains how to structure and code an IMS/VS application program, and it gives some suggestions on application testing and documentation. The third part of the manual contains reference information on application programming. Readers are expected to know COBOL, PL/I, or assembler language.

IMS/VS Application Programming for CICS/VS Users

This manual provides information on writing DL/I application programs that will run under CICS/VS. It explains how to use both command level programming and the DL/I call interface to access DL/I data bases. Readers are expected to know COBOL, PL/I, or assembler language, and are also expected to be familiar with CICS/VS application programming concepts and techniques.

IMS/VS Application Programming Reference Summary

This is a reference card for experienced application programmers. The card summarizes information contained in the reference section of *IMS/VS Application Programming*, and includes information on:

- DL/I call formats
- Data communication call formats
- Segment search argument structure and usage
- Command code meanings and usage
- Status codes

Administering IMS/VS Data Bases and the IMS/VS System

IMS/VS Data Base Administration Guide

This manual is a guide to the tasks involved in administering IMS/VS data bases. These tasks include designing, implementing, loading, monitoring, tuning, and modifying IMS/VS data bases. The manual also discusses participating in design reviews, analyzing data requirements, establishing security, and setting up standards and procedures. Readers should have a basic understanding of IMS/VS and application programming, and of the operating systems and access methods used at their installations.

IMS/VS System Administration Guide

This manual describes the tasks involved in administering an IMS/VS system. The tasks it describes include designing, implementing, monitoring, tuning, testing, and modifying an IMS/VS DB/DC system. It also discusses establishing IMS/VS security, and considerations in administering multiple IMS/VS systems and IMS/VS systems that share data. Readers should have a good understanding of IMS/VS application programming and DL/I data bases, and should be familiar with the IMS/VS online system and the operating system and access methods used at their installations.

IMS/VS Message Format Service User's Guide

This manual describes how to define and use Message Format Service (MFS). It discusses designing MFS applications and administering MFS application systems as well as defining the MFS control blocks.

People who use this manual should be familiar with IMS/VS application programming and message processing, and with the operating characteristics of the terminals they will use with IMS/VS.

IMS/VS Utilities Reference Manual

This manual is a reference book for people who use the IMS/VS utilities to administer data bases and the IMS/VS system. It describes each of the IMS/VS system utility programs and explains how to execute the utilities. The sections in this manual explain how you use the IMS/VS utilities for:

- Defining and maintaining DL/I data bases and PSBs
- Allocating, monitoring, and recovering the IMS/VS logs
- Performance reporting and service
- Defining and maintaining MSDBs and DEDBs

People who use this manual should understand IMS/VS concepts and facilities, OS/VS and OS/VS generation, data communications, and the access methods used by IMS/VS.

IMS/VS System Programming Reference Manual

Used with the *IMS/VS Installation Guide*, this manual provides the information necessary to install, customize, and maintain the IMS/VS system:

- The IMS/VS procedure library and execution parameters

- Exit routines written by IMS/VS users

- IMS/VS storage estimates

It also contains information on IMS/VS support for the IBM System/3 and the IBM System/7. Readers should be familiar with IMS/VS concepts and facilities, OS/VS and OS/VS generation, data communications, and the access methods used by IMS/VS.

IMS/VS Programming Guide for Remote SNA Systems

This manual explains the IMS/VS support for programmable logical units operating in an SNA environment, including the IBM 3600 and 4700 Finance Communication Systems, and the IBM 3790 Communication System. It addresses the areas with which programmers and analysts involved in communicating with IMS/VS must be familiar, and describes IMS/VS support for Intersystem Communication of the Multiple Systems Coupling function.

This manual is for system programmers of both the IMS/VS system and the supported logical units. Readers are expected to have a knowledge of IMS/VS, particularly of the Data Communication Feature, and should be familiar with the Systems Network Architecture (SNA) and VTAM concepts and facilities that govern communication between a VTAM application program and a VTAM logical unit.

Developing Operating Procedures and Operating the IMS/VS System

IMS/VS Operations and Recovery Guide

This manual covers the tasks of:

- Determining which tools and options will be used in operating IMS/VS

- Developing procedures for the master terminal operator (MTO)

- Developing procedures for the end user

This manual is intended for people who operate and recover IMS/VS. Readers should be familiar with the IMS/VS Data Base system, the Data Communications Feature, the utility programs, and Data Base Recovery Control.

IMS/VS Operator's Reference Manual

This manual contains reference information on the IMS/VS operator commands. It explains what the IMS/VS commands do and how to use them. This manual is for people who—using a terminal—start, control, use, and stop an IMS/VS system.

IMS/VS Data Base Recovery Control: Guide and Reference

This manual describes DBRC and includes information on these topics:

- General information on using DBRC

- Reference information needed for DBRC operation
- A detailed description of the elements that DBRC includes, and how they work
- Explanations of the DBRC commands

It is intended for people responsible for operating and recovering IMS/VS. Readers should have a good understanding of IMS/VS.

Servicing IMS/VS

The manuals listed below are intended to help diagnosticians identify problems with IMS/VS in as short a time as possible, and to use the IBM Support Center effectively.

IMS/VS Messages and Codes Reference Manual

This manual explains the reason the message was issued, describes the problem, system action, and programmer response, and helps diagnosticians to reinstate IMS/VS or the application program as soon as possible. Readers should have a good understanding of IMS/VS, and should be familiar with OS/VS and OS/VS system generation, data communications, and the access methods used by IMS/VS.

IMS/VS Failure Analysis Structure Tables (FAST) for Dump Analysis

If IMS/VS or an application program terminates abnormally, diagnosticians can use this manual to learn about the probable cause of the abnormal termination. This manual helps diagnosticians to analyze the problem by identifying conditions from data in registers as well as bit settings. This information can help them to determine the cause of the error.

IMS/VS Diagnostic Aids

The *IMS/VS Diagnostic Aids* manual provides product-specific procedures to locate basic facts about a problem. This manual will help you develop the keyword description of a problem, and instruct you how to use the keywords as search arguments in the software support data base. The manual identifies the information you need to begin your dialog with the Level 1 or Level 2 service representative.

If it becomes necessary for you to submit an APAR for a problem, the *Diagnostic Aids* manual contains instructions about how to prepare the documentation that must accompany the APAR. The appendixes contain cross-reference information needed to follow the diagnostic procedures. Readers should have a thorough understanding of IMS/VS.

IMS/VS Program Logic Manual

The *IMS/VS Program Logic Manual* provides a summary of the major functions of IMS/VS and a brief overview of its logic and organization. The PLM is organized to provide entry points into the logic descriptions using the facts you gathered in the diagnostic procedures and the cross-reference tables in the *Diagnostic Aids* manual. The PLM has six sections. Its Introduction describes how to use the information contained in it.

The *IMS/VS Program Logic Manual* also provides assistance in interpreting storage dumps and in using diagnostic tools supplied with the product. The manual supports the dialog between the diagnostician and the Level 2 representative to investigate new problems.

IMS/VS Data Base Recovery Control: Logic

This manual is a reference publication similar to the *IMS/VS Program Logic Manual* for problems concerning DBRC. Readers should have a thorough understanding of both IMS/VS and DBRC.

General Reference

IMS/VS Master Index and Glossary

Made available with the Data Base system, this manual is a consolidation of the indexes of all manuals in the IMS/VS library. It also contains a glossary of IMS/VS terms. The manual is intended to be a general reference and index for all IMS/VS users.

Additional Reference Information

You may also want to read the following publications to find out more about the way in which IMS/VS interacts with other subsystems:

- *CICS/VS General Information Manual*, GC33-0155
- *Batch Terminal Simulator II Program Description/Operations Manual*, SH20-1844
- *Application Development Facility General Information Manual*, GB21-9869

Appendix A. Highlights of IMS/VS Releases

This appendix lists the changes made to IMS/VS in this release and previous releases.

Release 3

New Product Packaging

The following are no longer separate features but are part of the IMS/VS system:

- **Fast Path** and **Multiple Systems Coupling** are a part of the IMS/VS Data Communication (DC) feature.

- **Data Base Recovery Control (DBRC)** is now a part of the IMS/VS Data Base system.

New Programming Facilities

System Logging

In an online system, log records are stored on direct access storage, instead of being written directly to tape. Logging to direct access storage can make the recovery of your system simpler and reduce operator involvement.

For an online system, log records are written to multiple online log data sets (OLDS). When an active online log data set becomes full, IMS/VS closes it and continues logging on the next OLDS. The IMS/VS-provided Log Archive utility transfers archival information to the system log data set (SLDS) automatically at user-specified intervals. Batch DL/I systems may also log to direct access storage.

Applications that require log tape input and the log utilities from previous releases of IMS/VS will continue to operate against the system log data set.

Online Change

IMS/VS permits the following resources to be added, changed, or deleted online without bringing down IMS/VS:

- Data base definitions and directories
- Application definitions and descriptors
- Transaction definitions
- MFS formats
- Fast Path routing code definitions
- Security definitions

The ability to make online modifications makes the system more adaptable to changing needs and allows longer periods of continued uninterrupted operation.

Enhancements to DEDBs

This release includes major changes to the use and capabilities of Data Entry Data Bases:

- Extensions to the DEDB Structure

  The DEDB structure has been expanded. A DEDB can contain up to 127 segment types, and up to 15 hierarchic levels. Application programmers can use up to 15 segment search arguments in their calls to the data base. The 127 segment types can include one sequential dependent segment type, if desired.

- Additionally, the call structure is expanded to include multiple segment search arguments (SSAs), command codes, and Boolean operators.

- Subset pointers

  You can use subset pointers with DEDBs to process long chains of segment occurrences more efficiently. Subset pointers allow you to more directly access a segment within a chain of segment occurrences. Several command codes are provided to allow you to read and update subset pointers from within an application program.

- Multiple Copies of Area Data Sets

  It is possible to use multiple copies of a DEDB area data set. One of the advantages of this is the protection it gives against data base failures; if there is an error in one area, and a copy of that area exists, application programs may continue processing the data in that area. Installations may create new copies of an area online, without stopping the area.

- Physical Child Last Pointer for Performance

  Installations can specify that they want new occurrences of unkeyed segments to be inserted at the end of a sequence of unkeyed segments. DEDBs use a physical child last pointer to make this insertion efficient.

- Record Deactivation for Data Base Availability

  If an I/O error occurs, any program trying to read the block of the data base in error (the Control Interval, or CI) will receive a status code indicating the error condition in that block. The application program can then continue with other processing as long as the data base hasn't been stopped. Record deactivation is automatic and the master terminal operator is informed. If you use multiple copies of the area in error, application programs may continue to access that data through those copies.

- DEDB Data Sharing

  Area level data sharing is supported for DEDBs; block level data sharing is supported for DEDBs that do not have a sequential dependent segment. Fast Path application programs and IMS/VS online application programs in separate IMS/VS systems can share DEDBs; IMS/VS batch programs cannot access DEDBs.

Data Base Recovery Control

Effective with this release, DBRC is required for your online system. In addition to the functions provided in previous releases of IMS/VS, several enhancements have been made, including:

- You can use a subset of the full DBRC function. For example, if you do not use data sharing, you can use DBRC to control logging only.
- DBRC has its own address space.
- You can dynamically allocate RECON data sets and the use of RECON data sets is continuous, despite I/O errors
- DBRC generates the JCL required to run the Log Archive and Log Recovery utilities.

DBRC also monitors the status of the online and system logs used by the IMS/VS online system and, if recovery is necessary, determines those log volumes that should be used as input to the recovery process.

System Definition Performance Improvements

IMS/VS 1.3 provides an optional preprocessor that can improve the performance of stage 1 system definition, and reduce the need for multiple stage 1 runs. The preprocessor performs the following checks of stage 1 source statement input:

- It checks resource names, within resource type, to ensure that no duplicate names exist.
- It cross-checks names of transactions, multiple systems, and logical terminals against each other to determine that there are no duplicate names among these.
- It checks resource names to ensure that they are alphabetic and of the appropriate length.

Additionally, a new, improved sort algorithm has been provided. You may specify that sorting be performed in either stage 1 or stage 2 of system definition.

Increased System Definition Limits

In previous releases, specification of each of the resources named below was limited to the inclusion of 5000 macro statements defining that resource. That limit has been removed. Now, the only limit on the number of these resources that may be defined is that imposed by your machine resources and the limits of your operating system.

- Application programs
- Transaction codes
- System/3 or System/7 station addresses
- Groupings of VTAM terminals (of the same type)
- Physical terminals
- Logical terminal names
- Subpools
- Routing codes

The limit on the number of data bases that may be defined is increased from 5000 to 32,700.

255 User Dependent Regions

If your operating system is OS/VS2 MVS, the number of concurrent user dependent regions and Fast Path output threads that may be defined has been increased from 31 to 255.

IRLM Enhancements

The IRLM has been extended to provide local locking in addition to global locking. (If the IRLM isn't included in your system, the PI lock manager is still used for locking services.)

The following additional enhancements have been made:

- There are no longer any IRLM restrictions on naming an IRLM VTAM application. (This enhancement also applies to IMS/VS Version 1 Release 2.)
- The IRLM control block structure can reside in its own address space.
- A trace can be started at IRLM initialization.

Message Format Service Enhancements

MFS has been enhanced to support users of MVS/XA; the MFS buffer pool has been moved from the IMS/VS private area to the IMS/VS extended area. Other enhancements include:

- The use of concatenated format data sets to allow concurrent input/output processing.
- An online dynamic MFS directory
- Increased maximum size for the MFS format pool, and a larger maximum number of fetch requests (for users of MVS/XA and MVS/370).

DL/I Subordinate Address Space

DL/I-related code, control blocks, and buffers can be contained in a separate address space. Use of the DL/I subordinate address space reduces the CSA storage requirements of the IMS/VS control program.

New Products and Devices

Network Terminal Option (NTO) Device Support

IMS/VS supports the use of the IBM Network Terminal Option Program Product. NTO provides a non-SNA start/stop terminal interface to VTAM for 3101, 2740, 2741, TTY, and TTY-compatible devices.

3290 Information Panel

IMS/VS supports the IBM 3290 Information Panel, which has a maximum screen size of 9920 bytes. (The maximum screen size supported in previous releases was 4K (4096) bytes.)

4700 Finance Communication System

The IBM 4700 Finance Communication System is supported by IMS/VS in the same way as is the IBM 3600 Finance Communication System.

CICS/VS-DL/I Primer Sample Programs

Code is provided to support the CICS/VS-DL/I Primer sample programs. This code has been integrated with the IMS/VS Primer code from previous releases of IMS/VS and will be distributed as a part of the IMS/VS Data Base System.

Release 2

New Programming Facilities

- IMS/VS can be generated as both a portable and nonportable system. A portable system is one that is generated using MVS/System Product Release 3. It will run on either System/370s or System/370s operating in Extended Architecture mode.

  A nonportable system is one that is generated using MVS/System Product Version 2 Release 1. This IMS/VS system will run only on System/370s operating in Extended Architecture mode.

- Data sharing provides control for two or more IMS/VS systems to access DL/I data bases concurrently. The IMS/VS systems can be in one processor, or they can be in separate processors. IMS/VS systems can share data on two levels: the data base level and the block level. At the data base level, application programs in each IMS/VS system can update the data, while others only read it. At the block level, several IMS/VS systems can update the data concurrently.

- The local storage option is able to use cross memory support in MVS/System Product Release 2. Cross memory support eliminates the need to switch from a dependent region task control block (TCB) to a control region TCB during parallel DL/I processing.

- For printers that use SLU 1 SNA Character Set Data Streams (SCS1), MFS makes it possible to set the vertical format on a message by message basis; to set the line density without losing a print line; and to use the extended attributes of color, highlighting, and programmed symbols.

- For printers that are defined as 3270P, MFS makes it possible to use the extended attributes of color, highlighting, and programmed symbols.

New Products and Devices

Devices and features supported by this release of IMS/VS are as follows:

- Support of the IBM 3604 Administrative Keyboard Display Model 7 extends the IMS/VS support of 3600 displays to include the 3604 Model 7 with its larger screen size (24 x 80). The IMS/VS support is the same regardless of the particular 3600 display attached.

- Support of the IBM 3375 device expands the OSAM device support to include Count-Key-Data (CKD) support of the 3375 Storage Facility. IMS/VS also supports the 3375 through VSAM in all data formats. The 3375 Direct Access Storage Device may be used for data bases, message queues, scratchpad areas (SPAs), and for restart data sets using VSAM and OSAM.

- Support of the IBM 3380 Direct Access Storage Device expands the OSAM device support. The high-speed data transfer 3380 may be connected to standard I/O channels through the speed matching buffer feature 6650 on the 3880 control unit. The 3380 Direct Access Storage Device may be used for data bases, message queues, scratchpad areas (SPAs), and for restart data sets using VSAM and OSAM.

Release 1.6

New Programming Facilities

Support is provided for Intersystem Communication as a function of the Multiple Systems Coupling (MSC) feature. Intersystem Communication permits multiple communication sessions between IMS/VS and other subsystems, such as CICS/VS, a user-written system, or another IMS/VS system.

A new connection type using the logical unit type 6 (LU 6) protocol is available for IMS/VS-to-IMS/VS MSC sessions. This link is referred to as MSC VTAM.

Support of secondary logical unit type 4 (LU 4), which is functionally equivalent to SLU 1 for comparable components, provides applications (through MFS) with device independence for consoles, printers, and card components. Data transparency is provided for media components.

MSC Directed Routing extends MSC by allowing an application program to specify the multiple system name (MSNAME) and destination within that system for a message to another system. The application program can receive the MSNAME of the system that originally scheduled it.

Application programs may, optionally, bypass MFS editing or basic editing when communicating via 3270 or SLU 2 devices. The user can leave the screen in an unprotected mode which allows IMS/VS to send output to the device without requiring input from the device. Optionally, the application program can also control the locking and unlocking of the keyboard.

With Extended Graphic Character Sets, the user can have more than one character set at a time available for use with the 3270 display system.

A new parameter which allows the user to select the MVS local storage option is available to the IMS procedure. This option permits the user to load some IMS/VS modules and buffers into the private storage area of the IMS/VS control region that were previously loaded in the Common Storage Area (CSA).

The user can run up to 31 dependent regions concurrently active in an online IMS/VS system. This support is within the constraints of the operating system or subsystem being used. The Fast Path feature also supports 31 dependent regions and up to 31 output threads.

IMS/VS Release 1.6 is installable with SMP Release 4 and defines IMS/VS distribution and system libraries in an SMP format. SMP can support multiple IMS/VS nucleuses in one RESLIB and multiple IMS/VS systems from a single set of distribution libraries.

New Products and Devices

Devices and features supported by this release of IMS/VS are as follows:

- IBM 3274 Model 51 Controller
- IBM 3274 Model 52C Controller with Extended Graphic Character Set support
- IBM 3278 Model 52 Display with Extended Graphic Character Set support
- IBM 3279 Color Display Station
- IBM 3283 Model 52 Printer with Extended Graphic Character Set support
- System/34 with SLU 2 support in addition to SLUs 1 and P
- System/38 which uses SLU 1 support
- IBM 3600 (Via the 3600 Administrative Application Support program product) using SLU 2 support.
- IBM 3650 Retail Store System, which uses SLU P support
- IBM 3680 Programmable Store System, which uses SLU P support
- IBM 3730 Distributed Office Communication System, which uses SLU P support
- IBM 5520 Administrative System, which uses SLU P support
- IBM 303X series processors
- IBM 4300 series processors
- IBM 8100 Distributed Processing Programming Executive (DPPX) and Distributed Processing Control Executive (DPCX) accessed as IMS/VS SLU 1, 2, and P support.
- IBM 6670 Information Distributor which uses SLU 4 support

Appendix B. System Configuration and System Control Programs

This appendix describes the requirements for an IMS/VS system configuration and lists the software support required to run IMS/VS.

System Configuration

Processors

IMS/VS executes under the virtual storage operating systems (OS/VS1 and OS/VS2) in System/370 models 138, 145, 148, 155II, 158, 165II, and 168, and the 303X, 3081, and 4300 processors.

With MVS/Extended Architecture (MVS/XA), IMS/VS executes on the IBM 3081 Processor Complex operating in Extended Architecture mode, with the MVS/System Product Version 2 (MVS/SP V2), and the Data Facility Product (DFP) installed.

It is suggested that the Model 370/138 be limited to the IMS/VS Data Base System only.

System Console

OS/VS1 and OS/VS2 requirements apply.

Tape Units

At least one 2400 or 3400 9-track tape unit is required.

Direct Access

For system libraries and working storage space, any devices supported by the operating system are allowed. Estimates of minimum space for system use and maintenance are:

System	**3330 cyls**	**3350 cyls**	**3375 cyls**	**3380 cyls**
DB	270	164	222	140
DB/DC	457	274	371	222
DB/DC/MSC	479	285	382	228
DB/DC/Fast Path	534	318	431	255

These figures are for planning purposes only.

For IMS/VS data base storage, within the capabilities and restrictions of BSAM, QSAM, ISAM, and VSAM support, the following are allowed:

2314/2319[1]	Direct Access Storage Facility
2305[1]	Fixed-Head Storage
3330 Models 1,2,11	Disk Storage
3333 Models 1,11	Disk Storage
3340	Direct Access Storage
3344	Direct Access Storage
3350	Direct Access Storage
3375	Direct Access Storage
3380	Direct Access Storage
3850	Mass Storage System

Note:

[1] Extended Architecture does not support these devices.

Fast Path

Fast Path runs on the processing units supported by OS/VS2 MVS/SP Version 1 Release 3 and MVS/SP Version 2 Release 1. All devices supported by the operating system are allowed for system libraries and working storage. MSDBs are stored as DASD data sets when Fast Path is shut down. The space required for MSDBs depends on their number and size. Direct access devices supported by IMS/VS and VSAM improved control interval processing are available for DEDBs. Fast Path *does not* operate in an OS/VS1 environment.

Fast Path supports the following communication devices:

Terminal Type	**Polled Mode**	**Local Mode**	**SNA**
2740-1[1]	BTAM		
2740-2	BTAM		
3270 (BSC/local)	BTAM	BTAM	
3270 (BSC/local)	VTAM	VTAM	
3600[2]			VTAM
4700			VTAM
3767			VTAM
3770			VTAM
SLU 1[3]			VTAM
SLU 2			VTAM
SLU 4[3]			VTAM
SLU P			VTAM
LU 6			VTAM
NTO Devices			VTAM

Notes:

[1] Station control only.
[2] 3614 support via 3601 only.
[3] Must have at least one input component.

Multiple Systems

Each system in a multiple systems configuration must consist of an entire IMS/VS DB/DC product with its own IMS/VS master terminal. The IMS/VS systems in the configuration must be of compatible releases.

The following systems are compatible with each other:

- IMS/VS Version 1 Release 1.6

- IMS/VS Version 1 Release 2

- IMS/VS Version 1 Release 3

When the physical link is a binary synchronous communication line, BTAM is required. Point-to-point contention mode (BSC1) is supported.

When the physical link is a main storage to main storage link, OS/VS1 and OS/VS2 MVS are supported.

When the physical link is a channel-to-channel adapter, the System/370 channel-to-channel adaptor is required. This is a standard priced feature (#1850) on System/370 models 145, 155-II, and 158. For a model 165-II or 168, the adaptor is available as RPQ WE4259 for an IBM 2880 channel. A channel-to-channel link is supported only for OS/VS2 MVS.

When the connection is via VTAM, ACF/VTAM Version 1 Release 1 or later is required. The Synchronous Data Link Control (SDLC) line discipline is used. The parallel session capability is not available with ACF/VTAM Version 1 Release 1, or with ACF/TCAM.

If CICS/VS will participate with IMS/VS in an Intersystem Communication network, CICS/VS Version 1 Release 5 or later is required.

Data Sharing Requirements

- Data Base Recovery Control is required for data sharing.

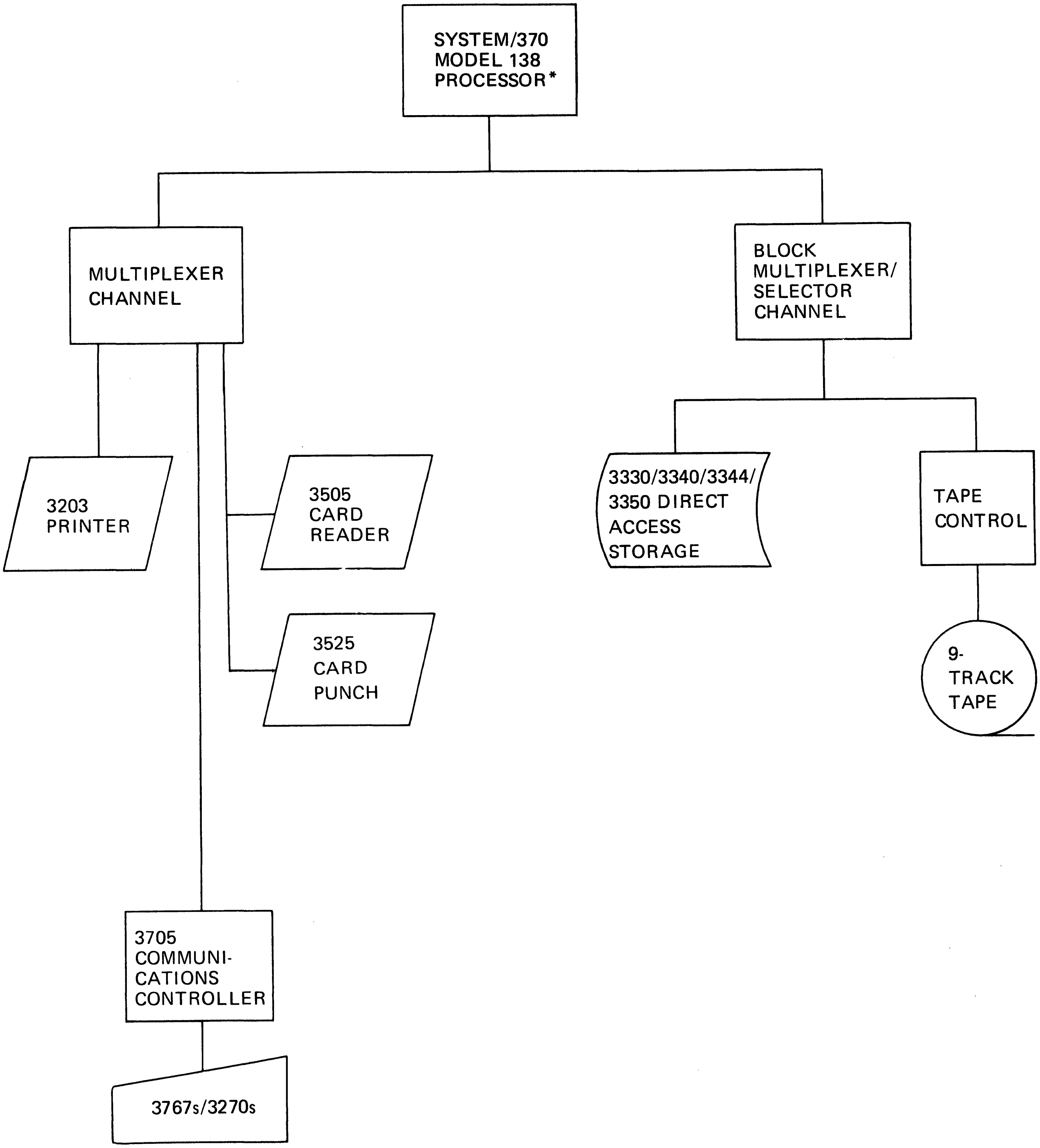
- For interprocessor sharing, either a 3705 controller or a channel-to-channel (CTC) link can be used. If a 3705 controller is used, ACF/VTAM Version 1 Release 2 Multisystem Networking Facility or a subsequent release with the appropriate ACF/NCP release is required. If a CTC link is used, ACF/VTAM Version 2 Release 1 is required.

- IMS/VS systems running under OS/VS1 or OS/VS2 may share data at the data base level. For block level sharing, OS/VS2 MVS is required; see the following.

- For block level sharing, the IMS/VS Resource Lock Manager (IRLM) is required. IRLM is a component of the IMS/VS Data Base System. IRLM must be defined as an OS/VS2 MVS subsystem and as an ACF/VTAM application in each processor.

Minimum IMS/VS Configurations

The minimum configurations for telecommunications and batch-only IMS/VS systems are shown in Figure 14 on page 73 and Figure 15 on page 74 .



*With appropriate features

Figure 14. Minimum Configuration—IMS/VS Telecommunications

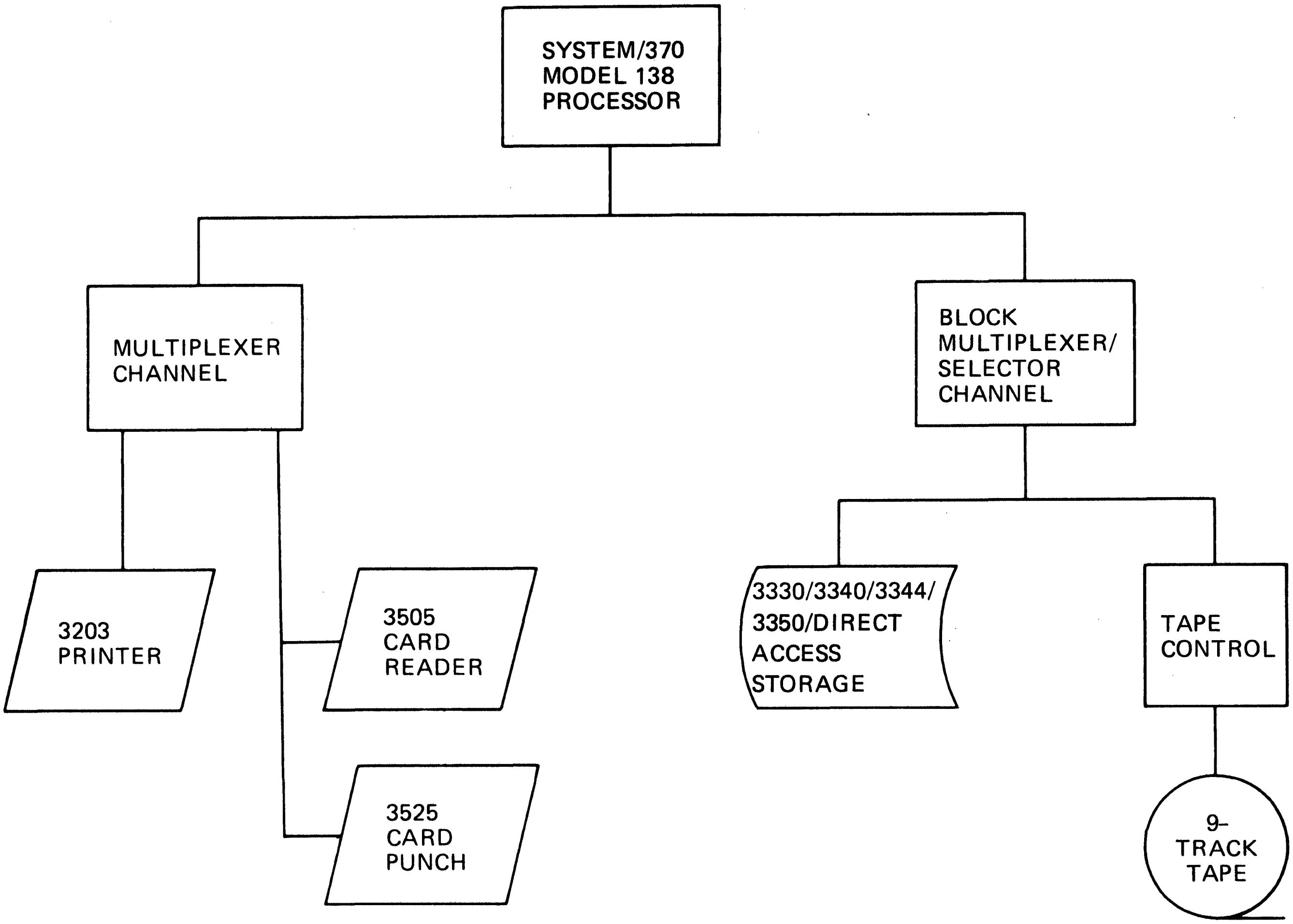


Figure 15. Minimum Configuration—IMS/VS Batch-Only

Sample IMS/VS Configuration

Figure 16 shows a sample IMS/VS configuration. The terminals attached to the 3705 would typically be some combination of 3767 terminals, 3270 terminals, and 3790/3600 systems.

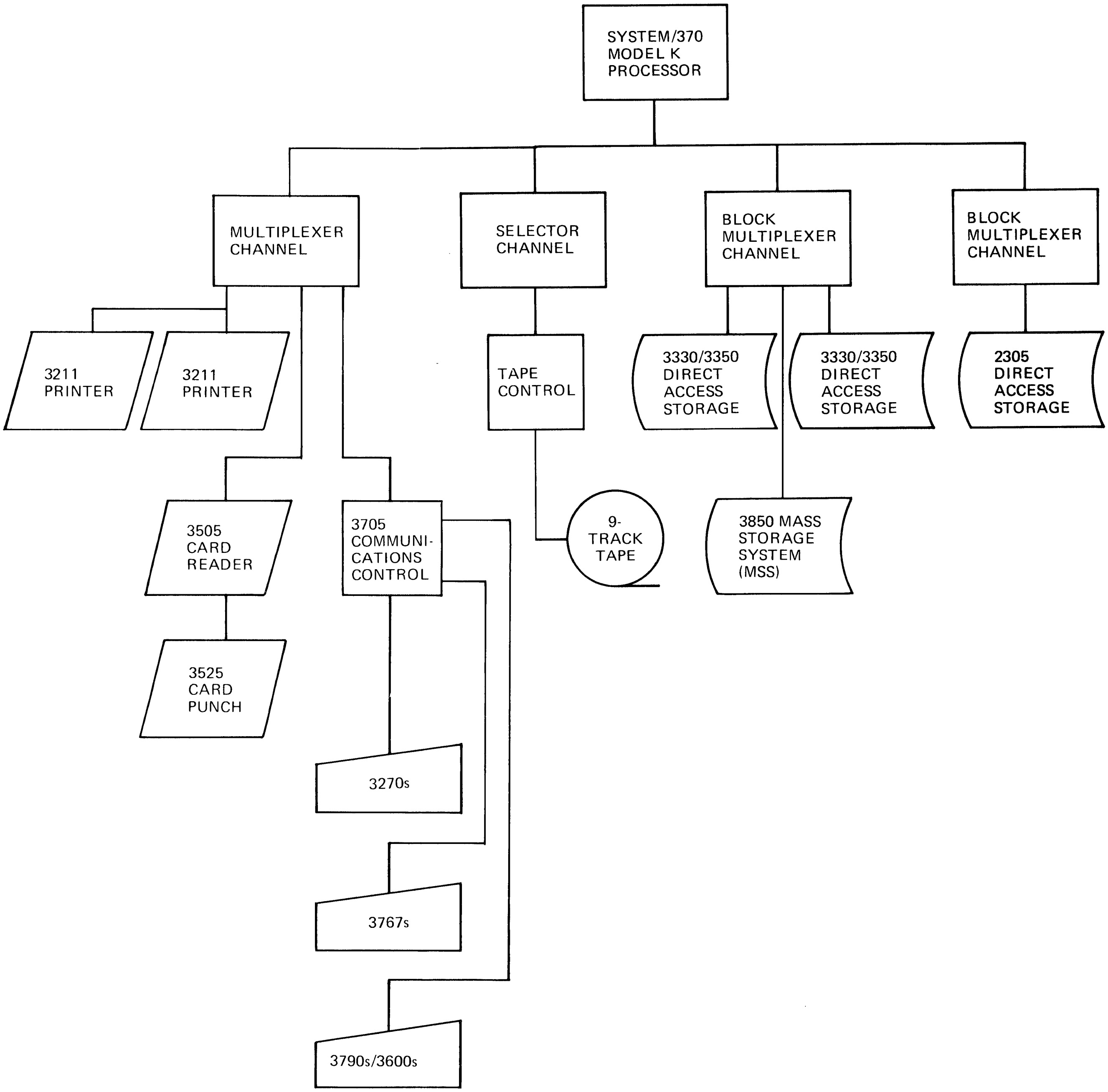

SYSTEM/370 MODEL K PROCESSOR
MULTIPLEXER CHANNEL
SELECTOR CHANNEL
BLOCK MULTIPLEXER CHANNEL
BLOCK MULTIPLEXER CHANNEL
3211 PRINTER
3211 PRINTER
TAPE CONTROL
3330/3350 DIRECT ACCESS STORAGE
3330/3350 DIRECT ACCESS STORAGE
2305 DIRECT ACCESS STORAGE
3505 CARD READER
3705 COMMUNI-CATIONS CONTROL
9-TRACK TAPE
3850 MASS STORAGE SYSTEM (MSS)
3525 CARD PUNCH
3270s
3767s
3790s/3600s

Figure 16. Sample Configuration—IMS/VS System

IMS/VS Storage Requirements

The *IMS/VS System Programming Reference Manual* contains in Chapter 5, "IMS/VS Storage Estimates" the detailed information necessary to estimate IMS/VS storage requirements.

For the storage requirements of OS/VS1 and OS/VS2, refer to *OS/VS1 Storage Estimates*, GC24-5094, or *OS/VS2 MVS Initialization and Tuning*, GC28-0681, and *OS/VS2 MVS Overview*, GC28-0984.

For information on BSAM, BTAM, VTAM, and VSAM, consult the appropriate storage estimates manual. For storage estimate information on GAM and ARAM, consult Chapter 5, "IMS/VS Storage Estimates " in the *IMS/VS System Programming Reference Manual.*

System Control Programs

Operating Systems

IMS/VS Version 1 Release 3 operates under the following virtual storage operating system configurations:

- OS/VS1 Release 7.0
  - Fast Path is *not* supported for the OS/VS1 operating system with this release of IMS/VS V1.
  - The IRLM is *not* supported for the OS/VS1 operating system.
- OS/VS2
  - For the System/370 environment (with their appropriate prerequisites):
    - MVS System Product-JES2 (5740-XYS) or -JES3 (5740-XYN) Version 1 Release 3 (referred to below as MVS/SP V1 R3)
    - Data Facilities Device Support Release 1 (5740-AM7)
    - Data Facilities Extended Function Release 1 (5740-XYQ)
  - For the MVS/Extended Architecture (MVS/XA) environment (with their appropriate prerequisites):
    - MVS System Product-JES2 (5740-XC6) or -JES3 (5665-291) Version 2 Release 1 (referred to below as MVS/SP V2 R1)
    - MVS/XA Data Facility Product Release 1 (5665-284; referred to below as MVS/XA DFP R1)

For proper execution of IMS/VS, system definition and system execution must be performed under the same operating system release. Unless required by the operating system, a new IMS/VS system definition using the same IMS/VS release does not make it necessary to recompile user application programs.

The selection of an OS/VS configuration has some effect on the potential performance, reliability, and availability characteristics of IMS/VS. Certain options are required by IMS/VS in all of the applicable configurations.

IMS/VS is designed to be a **portable** system. That is, when generated using OS/VS2 MVS/System Product Release 3, it can run on either a System/370 or a System/370 running in Extended Architecture mode.

IMS/VS Version 1 Release 3 can also be generated as a **nonportable** system. That is, when generated using OS/VS2 MVS/XA, it can run only on a System/370 running in Extended Architecture mode. MVS/XA comprises MVS/System Product Version 2 Release 1 and the Data Facility Product. If a nonportable system (one using OS/VS2 MVS/XA) is generated, Assembler H Version 2 Release 1 is also required.

An IMS/VS Data Base System operates only in virtual=virtual storage. In a DB/DC system running under OS/VS1 or OS/VS2 MVS, the control, message processing, and batch message regions operate only in virtual=virtual (V=V) storage.

When the IMS/VS control region or partition is run in a virtual environment, the user is given the ability to fix specific portions of the control program. This permits tuning of the IMS/VS system to a particular environment.

IMS/VS also operates in a VM/370 virtual machine under control of OS/VS1 or OS/VS2 MVS with the following restrictions:

- If binary synchronous terminals through BTAM are used, communication line timeouts should be expected. Note that this timeout of terminals is not limited to VM/370, but when it does occur, it affects recovery because polling is inconsistent when BTAM operates under VM/370.

- The IMS/VS statistics utility is not supported.

- If problems are encountered when operating IMS/VS under VM/370, it may be advantageous to re-create the problem in a stand-alone OS/VS environment.

IMS/VS operation in a VM/370 environment is intended primarily for use in program development and testing. IMS/VS can operate under VM/370 for production purposes. If you have specific throughput or terminal response time requirements, you should plan to benchmark under VM/370 to ensure that the proposed configuration will meet your needs. Consideration should be given to making the virtual machine running IMS/VS a favored virtual machine.

Programming Language

IMS/VS is written in assembler and PL/S. Those modules of IMS/VS Version 1 either written partially or completely in PL/S (DBRC, IRLM, and Fast Path) are available only in object code.

Access Methods

An IMS/VS Data Base System uses Basic Sequential Access Method (BSAM), Queued Sequential Access Method (QSAM), Indexed Sequential Access Method (ISAM), and/or Virtual Storage Access Method (VSAM). IMS/VS uses VSAM shared resources support, which reduces main storage requirements by sharing buffers and control blocks across VSAM data sets.

An IMS/VS DB/DC System uses, in addition, Basic Telecommunications Access Method (BTAM) and Virtual Telecommunications Access Method (ACF/VTAM). If local support for IBM 2260 Display Stations is required, the Graphic Access Method and Graphic Programming Services are required. IMS/VS Version 1, Release 1.6 or later requires ACF/VTAM Release 1, 2, or 3 for VTAM support. If MSC or ISC will utilize parallel sessions or negotiable session initialization parameters (BIND), ACF/VTAM Release 2 or later is required.

Appendix C. Terminals Supported by IMS/VS

This Appendix lists the terminals that IMS/VS supports.

IMS/VS-Supported Product	Compatible Product	Contention Mode	Switched Mode	Polled Mode	Local Mode	SNA	Notes
1050			BTAM	BTAM			
2260-1 and 2				BTAM	GAM		
2265-1				BTAM			
2740-1	3767	BTAM	BTAM	BTAM			1
2740-2	(2740 mdls only)			BTAM			1,2
2741	CMCST	BTAM	BTAM				1,3
2770	3770(BSC), 5110			BTAM			
2780		BTAM		BTAM			
2980				BTAM			4
3270			BTAM	B/V	B/V	VTAM	1,2
3600	4700					VTAM	1,2,5,6
3614						VTAM	4,5
3740	5280		BTAM				4,7
3767						VTAM	1,6
3770						VTAM	1,6
SLU 1 (i.e., 3230, 3232, 3262, 3287, 3767, 3268, 3770, 3770P, 3790 (type 2 batch & bulk print), 4700, 5280, S/32, S/34, S/38, 8100)						VTAM	1,2

Figure 17 (Part 1 of 2). Summary of Terminals Supported by IMS/VS

IMS/VS-Supported Product	Compatible Product	Contention Mode	Switched Mode	Polled Mode	Local Mode	SNA	Notes
SLU 2 (i.e., 3276, 3278, 3279, 3790 (3270 DSC feat.), 3600 Admin PP, 4700, 5280, 5520, 8100, 8775, S/34, Display writer)						VTAM	1,2
SLU 4 (i.e., 6670)						VTAM	1,2
SLU P (i.e., 3600, 3630, 3650, 3680, 3770PC, 3790, 4700, 5520, 8100, S/34, Series/1)						VTAM	1,2,5,8
LU 6						VTAM	1,2
NTO (i.e., 33/35, TTY, 2740, 2741, 3101, 3232, 3767, S/23)						VTAM	1,2
7770-3			ARAM				3,4
33/35, TTY	3101, 3232, S/23		BTAM				3
System/3				BTAM			8
System/7		BTAM		BTAM			8
System console					WTO(R)		
Card reader					BSAM		
Printer					BSAM		
Magnetic tape					BSAM		
DASD devices					BSAM		

Figure 17 (Part 2 of 2). Summary of Terminals Supported by IMS/VS

Notes to Figure 17 on page 79:

Key

ARAM	IMS/VS Audio Response Access Method
BSAM	Basic Sequential Access Method
BTAM	Basic Telecommunications Access Method.
B/V	BTAM or VTAM.
GAM	Graphic Access Method. Graphic programming service is required to receive GAM.
SNA	Systems Network Architecture. IMS/VS is independent of physical connection types used by VTAM for SNA.
VTAM	Virtual Telecommunications Access Method.
WTO(R)	OS/VS write-to-operator/write-to-operator-with-reply (WTO/WTOR) capability.

1. The IMS/VS Message Format Service (MFS) is available for this device. MFS editing can be bypassed on a message-by-message basis.

2. IMS/VS Fast Path allows the use of the compatible terminals.

3. IBM does not provide error detection capability for this device. Therefore, only inquiry-only transactions are allowed, even though the noninquiry capability exists.

4. User-written exit routines are required for the 3614, 3741, and 7770-3 terminals, and may be required in some environments for the 2890.

5. Although IMS/VS provides sample code for this terminal, additional user coding is required.

6. These terminals can be defined by identification number and as SLUs. The support for the multiple definition is not necessarily the same.

7. Connection between IMS/VS and this terminal requires a manual dial from the host.

8. IMS/VS provides no device-resident code for this device. Additional user coding is required to attach it to IMS/VS.

Glossary

The terms below represent only some of the more basic IMS/VS terms and acronyms used in this book. The *IMS/VS Master Index and Glossary*, described in Chapter 8, "IMS/VS Publications" on page 55, contains a more complete set of terms.

AOI. Automated Operator Interface, which allows installations to monitor and control IMS/VS activities.

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

batch-oriented BMP. A batch message processing program that does not access the message queue for input. It runs online and can send its output to an OS/VS output device.

BMP. Batch message processing, an IMS/VS batch processing program that has access to online data bases and message queues. BMPs run online, but like programs in a batch environment, they are started with job control language (JCL).

checkpoint. A time when significant system information is written on the system log and, optionally, the system is shut down. The information written to the log is primarily used to restart IMS/VS.

child segment. In a data base, any segment that is dependent on another segment above it (its parent) in the hierarchy. A child segment has only one parent segment.

command. A request from a terminal or AO (automated operator) program to perform a specific IMS/VS service, such as altering system resources or displaying specific system information.

data base. A collection of interrelated or independent data items stored together without unnecessary redundancy, to serve one or more applications.

data base record. In a data base, a collection of segments that contains one occurrence of the root segment type and all of its dependent segments arranged in hierarchic sequence.

data dictionary. A centralized collection of information about data, such as meaning, relationships to other data, origin, use, and format. It is useful in helping company management, data base administrators (DBAs), systems analysts, and application programmers to effectively plan, control, and evaluate the collection, storage, and use of data.

data sharing. The concurrent access of IMS/VS data bases by two or more IMS/VS systems. The IMS/VS systems can be in one processor or in separate processors.

DBD. Data base description, which defines the characteristics of a data base, such as the data base's organization and access method, the segments and fields in a data base record, and the relationship between types of segments.

DBRC. Data Base Recovery Control, a facility included in the IMS/VS Data Base system to permit easier recovery of IMS/VS data bases. DBRC maintains information required for data base recoveries, generates recovery control statements, verifies recovery input, maintains a separate change log for data base data sets, and supports sharing of IMS/VS data bases by multiple IMS/VS subsystems.

DEDB. Data entry data base, which provides a high level of availability for, and efficient access to, large volumes of detailed data.

dependent segment. In a data base, a segment that relies on a higher level segment in the hierarchy for its full definition.

DL/I. Data Language/I, a complete data management facility in the IMS/VS system. DL/I is the IMS/VS data manipulation language, a common high-level interface between a user application and IMS/VS. DL/I is invoked from PL/I, COBOL, or Assembler Language applications by subroutine calls. DL/I lets the user define data structures, relate structures to the application, load structures, and reorganize structures.

EMH. Expedited message handling, which processes single-segment input and output messages and, by bypassing IMS/VS message handling, allows messages to be processed more quickly.

field. The area within a segment that is the smallest unit of data that can be referred to.

hierarchy. The tree-like arrangement of segments in a data base, beginning with the root segment and proceeding down to dependent segments. As many as 15 segment levels may be defined.

IRLM. IMS/VS Resource Lock Manager, which is an IMS/VS component that provides a locking facility for use by IMS/VS subsystems that share data at the block level in an MVS environment.

ISC. Intersystem Communication, an extension of the IMS/VS Multiple Systems Coupling facility that permits the connection of IMS/VS to another IMS/VS subsystem, to CICS/VS, or to a user-written subsystem, provided both subsystems use ISC.

LTERM. Logical terminal, which names a logical device that is associated with one or more physical terminals.

message. Data transmitted between any two terminals, application programs, or IMS/VS systems. Each message has one or more segments.

message-driven program. An IMS/VS application program that processes messages and uses expedited message handling (EMH).

message queue. The data set on which messages are queued before being processed by an application program or sent to a terminal.

message switch. A terminal input message directed to another terminal without being processed by a message processing program.

MFS. Message format service, an editing facility that allows application programs to deal with simple logical messages instead of device dependent data, thus simplifying the application development process.

MPP. Message processing program, an IMS/VS application program that is driven by transactions and has access to online IMS/VS data bases and message queues.

MSC. Multiple Systems Coupling, a facility that lets geographically dispersed IMS/VS systems communicate with each other.

MSDB. Main storage data base, which stores and provides online access to an installation's most frequently used data.

MTO. Master terminal operator. The MTO uses IMS/VS commands to operate IMS/VS.

OLDS. Online log data set, which is where log records are kept for an online system. Log records are a primary tool in recovering IMS/VS.

PSB. Program specification block, which controls the segments and fields that an application program can view and the operations that it can perform.

parent segment. In a data base, a segment that has one or more dependent segments (its children) hierarchically below it.

root segment. In a data base, the highest segment in the hierarchy. In a data base record, there is only one root segment, and all other segments are dependent segments.

segment. The unit of access in DL/I; for the DB (data base) system, the smallest amount of data that can be transferred by one DL/I operation.

segment occurrence. In a data base, a single specific segment.

segment type. In a data base, a user-defined category of data.

SLDS. System log data set, which contains archive log records.

SNA. Systems network architecture, a set of standard rules and procedures used for telecommunication. Also, an integrated structure of equipment and programs.

transaction. A specific set of input data that triggers the execution of a specific process or job. A transaction is a message destined for an application program.

transaction code. A 1- to 8-character alphameric code that invokes an IMS/VS message processing program.

transaction-oriented BMP. A batch message processing program that accesses message queues for its input and output and which can also process input from OS/VS files and can create OS/VS files.

twin segments. In a data base, multiple occurrences of the same segment type under a single parent. Root segments are also considered twins to each other.

Index

IMS/VS Version 1
General Information Manual
GH20-1260-12

Reader's Comment Form

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

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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

Please use pressure sensitive or other gummed tape to seal this form.

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ____________________

Previous TNL ____________________

Previous TNL ____________________

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

GH20-1260-12

Reader's Comment Form

Fold and tape

Please do not staple

Fold and tape

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and tape

Please do not staple

Fold and tape

IBM®

GH20-1260-12

IMS/VS Version 1 General Information Manual (File No. S370-50) Printed in U.S.A. GH20-1260-12

IBM®