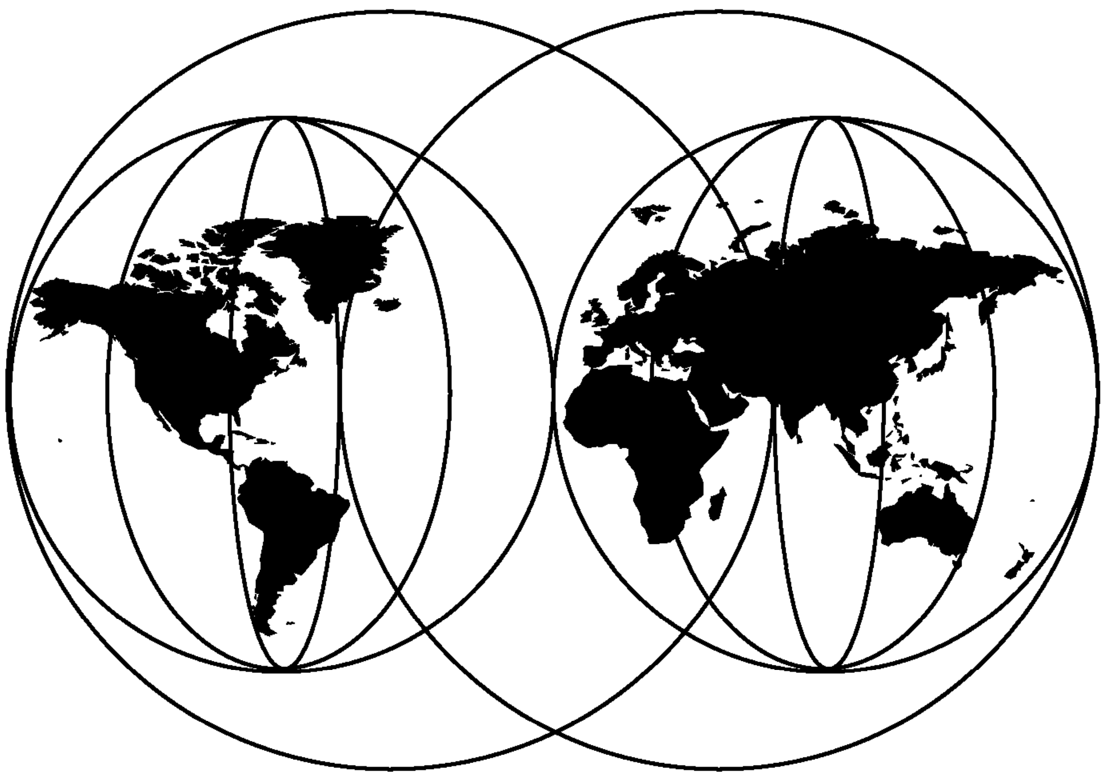
Interoperability Between VSE DL/I and OS/390 IMS DBCTL

Scott Chen, Nigel Miller, Cheng Jing, Li Jin Hao, Chen Shen



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5249-00

International Technical Support Organization

SG24-5249-00

Interoperability Between VSE DL/I and OS/390 IMS DBCTL

November 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix N, “Special Notices” on page 331.

First Edition (November 1998)

This edition applies to Version 6 of IMS/ESA, Program Number 5655-158, Version 1 Release 2 of CICS Transaction Server for OS/390, Program 5655-147, and Version 2 Release 3 of CICS/VSE, Program 5686-026, for use with the MVS/ESA or OS/390 and VSE/ESA Operating Systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures

Tables

Preface

This redbook can guide VSE customers as well as IBM representatives to an understanding of how the IBM CICS Transaction Server for OS/390 and the IMS/ESA DBCTL system work together to replace the VSE environment.

This redbook will help you migrate, tailor and configure the VSE application systems to the new OS/390 environment, step by step. The detailed explanation coupled with many sample procedures should make the process of migration less worrying and allow it to proceed more rapidly.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Scott Chen is a Data Management specialist at the International Technical Support Organization San Jose Center. He has over 25 years of experience in IBM large system and application development in the areas of MVS, DBMS, and documentation image management.

Nigel Miller is an IMS Specialist in the United Kingdom. He has 13 years of experience in various fields, including IMS, CICS, MVS, VSE, and VM. He has also worked on IDMS and ORACLE. He holds a degree in Chemistry from the Imperial College, London. He has worked at IBM for 3 years. His areas of expertise include IMS, CICS, and IDMS, which he has worked on worldwide.

Cheng Jing is a senior I/T Specialist of IBM China currently responsible for PRC S/390 Initiative Solution, the VSE to OS/390 migration project. She has 9 years experience in large systems and VSE-based products including CICS/VSE. She worked as an application programmer and system programmer before joining IBM China 5 years ago as a system engineer. She has experience as an S/390 senior I/T specialist responsible for large banking accounts, I/T solution design, and system installation.

Li Jin Hao is a senior Software Engineer and Manager in the Industry Commercial Bank of China, Hang Zhou Branch, China. He has 15 years of experience in the application development field. He holds a bachelor′s degree in computer science from Zhejiang University. His areas of expertise include online transaction development, software engineering, and application system design.

Chen Shen is a Software Engineer and Manager in the Industry Commercial Bank of China, Hang Zhou Branch, China. He has 10 years of experience in the application development field. He holds a master′s degree in business administration from Zhejiang University. His areas of expertise include online transaction development, software engineering, and application system design.

Thanks to the following people for their invaluable contributions to this project:

Bob Haimowitz
Vasilis Karras
Dave Bennin
Roy Costa

Fred Borchers
International Technical Support Organization, Poughkeepsie Center

Rich Lewis
IMS Advanced Technology Support Center
IBM US

Steve Zemblowski
Transactions Systems Support Center
IBM US

Dougie Lawson
IBM UK

Uwe Zumpe
VSE Project Office
IBM Germany

Thanks to Shirley Weinland Hentzell for editing the document.

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 347 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users	http://www.redbooks.ibm.com/
For IBM Intranet users	http://w3.itso.ibm.com/

- Send us a note at the following address:

  redbook@us.ibm.com

Chapter 1. Introduction

A great many changes are currently taking place in the S/390 processing world. Computer users have for several years been able to view their particular data processing environment as almost static, seeing new advances in technology as irrelevant to them. Thus, unless a particular application of new releases has immediate relevance, they see no need for any major change.

The Common European Currency is forcing many companies both in Europe and elsewhere into making major changes to all their financial applications. Substantial economic growth in many parts of the world is causing an enormous increase in the processing requirement. This is particularly true for many of the so-called *emerging market* countries. For many years, their isolation, both political and economic, from many of the world′s economic powers, meant that their slowly growing economic activity went unnoticed by the outside world. In addition, the political isolation also caused many of the technologically advanced countries, such as the United States, to restrict exports of high technology, such as IT equipment, to these countries. This further delayed their much needed modernization, keeping it below the level of the outside world. However, the collapse of the Soviet Union and the COMECON block forced many of these countries to open their door and accept assistance from world organizations such as IMF, World Bank, and UNDP. The result is an unprecedented economic boom during the last few years.

VSE applications are bound by the small and fixed amount of storage available to jobs (region size) which limits the amount of data they can handle at any one time. Such a limitation went unnoticed for years in these emerging market countries, because of low economic activity as well as isolation from countries with advanced technology. However, the increasing economic activity in recent years has forced many of these countries to start the search for other high-capacity IT tools to support their rapid growth. A combination of relaxing export restrictions in many industrial countries, such as the U.S., and higher capacity IT equipment causes many organizations within these countries to focus on converting their business systems to make use of this more recent technology and the operating systems that best exploit it. It is well known that the higher operating limits in a well-run OS/390 allow users to process data much faster. In addition, a great many more utilities are available for MVS, both from IBM and from other suppliers, which widen the choices that users have in deciding the best way to run applications.

Recent changes in both the types of machines available in VSE, and MVS operating systems have caused their ranges to converge, so that it now becomes possible to run VSE on much larger machines than before, and OS/390 releases of MVS can run on anything from a medium-sized workstation up to the very largest processors in the world. However, unlike VSE, OS/390 can now run many copies in parallel in what is known as a *sysplex*.

All these factors are converging to cause users with databases in database language 1 (DL/I databases) to want a more efficient way to access them, and to transfer to MVS where the range of utilities and on-line facilities are much greater. With the new flexible configurations available to MVS, users can now migrate their VSE images to MVS bit by bit, without needing any major redesign of application architecture. For example, users can therefore configure their

MVS images in whatever way best fits the machine they are running on, and have many more options.

1.1 The Old Way: Local DL/I

Users have been accustomed to accessing DL/I databases from a CICS region running either on VSE or MVS/ESA. In this approach, each CICS region controls its own set of databases in the same way as it would control its own VSAM files. Figure 1 illustrates a typical local DL/I environment.

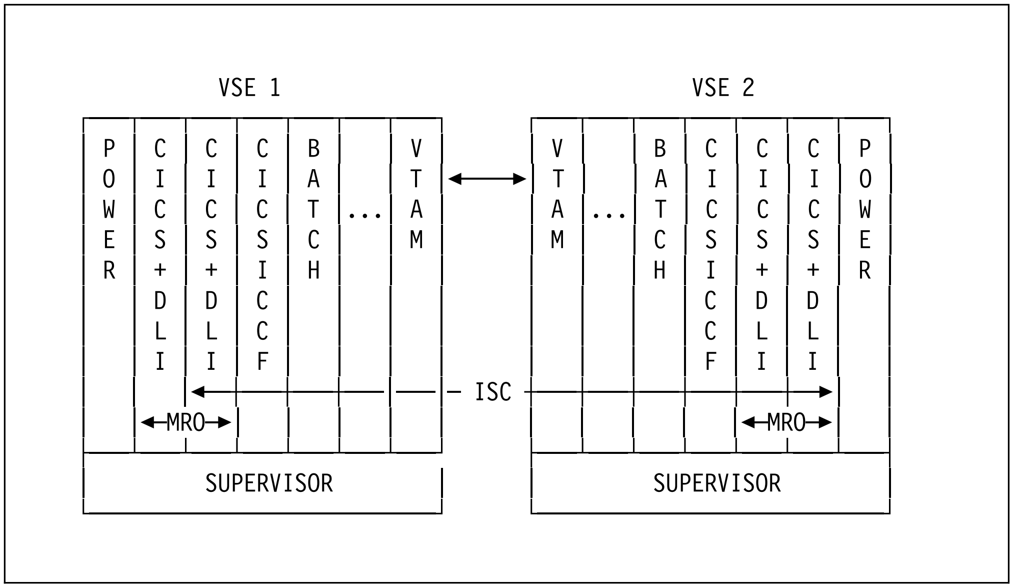


Figure 1. Typical Local DL/I Configuration

A CICS region needing access to another CICS region′s databases would access them through an ISC or MRO connection, defining those resources belonging to other CICS regions as *remote*. Such an environment would not only exist on a VSE platform but also for early versions of CICS on MVS platforms as well.

1.2 The New Way: DBCTL

With the new approach, many CICS regions can pool all their DL/I databases into one DBCTL region, which then acts as a central server to all the CICS regions that formerly owned all their own databases. Application programs do not normally need to be rewritten to use the new configuration. Figure 2 on page 3 shows a simple migrated DBCTL environment.

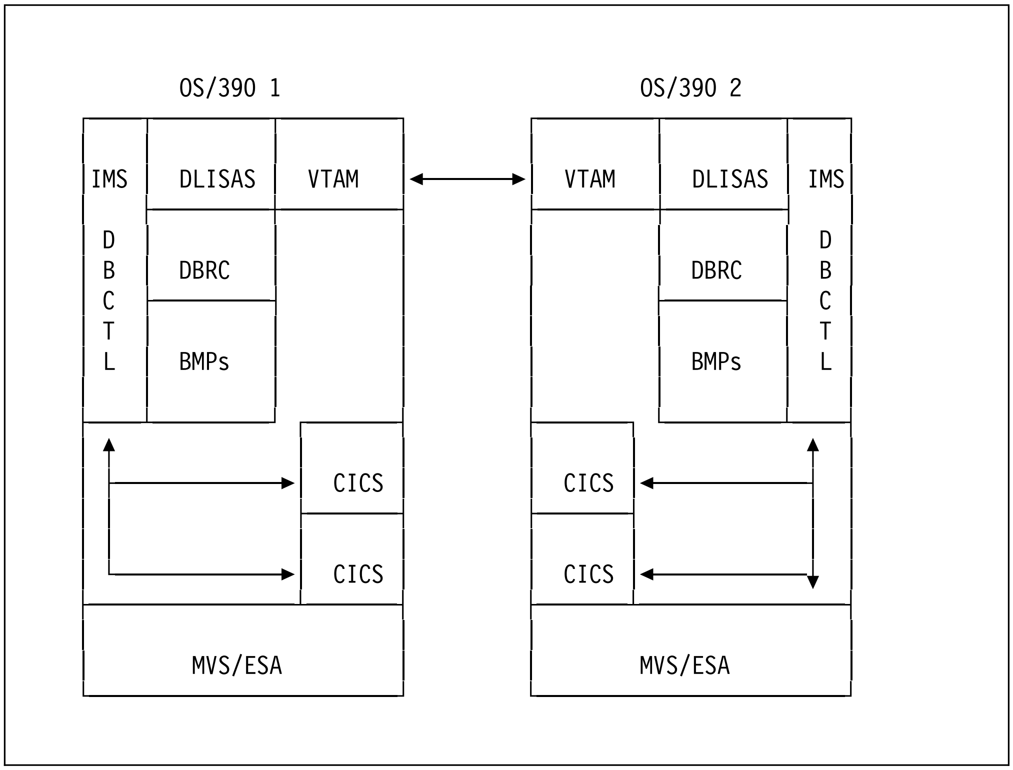


Figure 2. Migrated OS/390 IMS DBCTL Configuration

With further consolidation on applications and databases, the new CICS and DBCTL production environment could look like Figure 3.

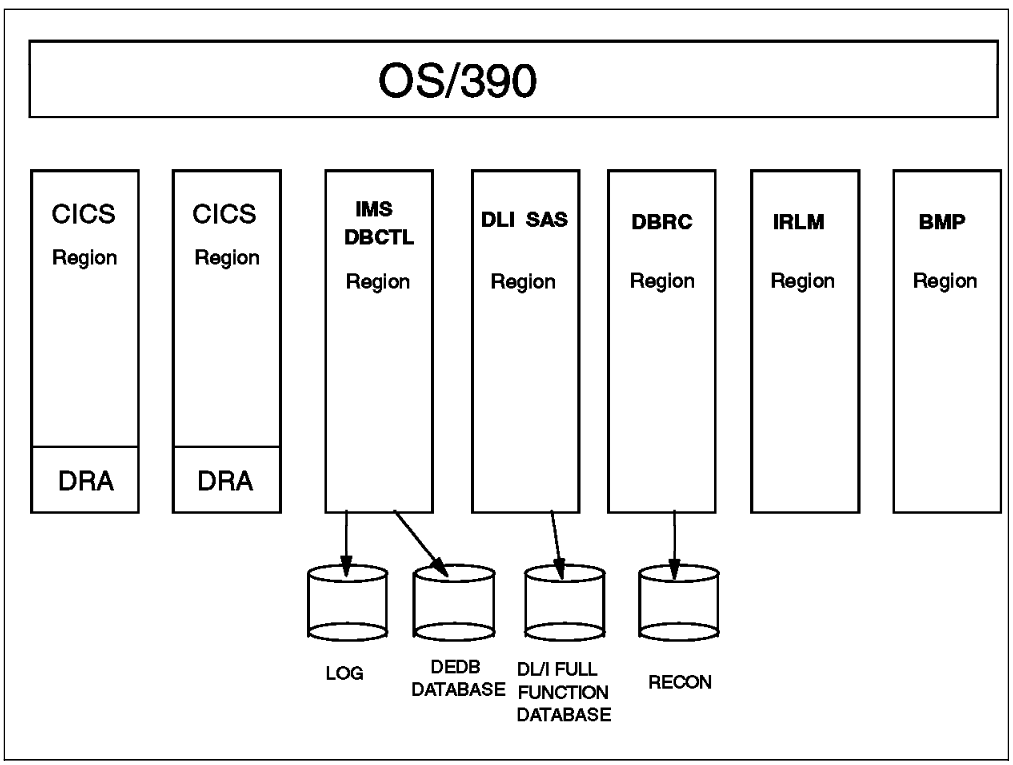


Figure 3. Sample OS/390 CICS and IMS DBCTL Production Configuration

1.3 What You Gain If You Convert

There are many advantages of conversion from CICS Local DL/I to DBCTL. For example:

- Access to databases becomes more efficient.
- Batch-DL/I jobs can be converted to BMPs, thus removing CICS storage contention. Conversion also enables users to use system service requests, such as symbolic checkpoint (CHKP) and extended restart (XRST), and to access GSAM databases. With BMPs, all logging goes to a single log (the IMS log), which eliminates the need for separate batch logs.
- CICS can be closed when on-line access to databases are not required.
- Database contention does not stop all CICS work.
- Applications can use IMS features such as Fast Path DEDB. Using data entry databases (DEDBs) enables users to have very large databases with high availability. DEDBs are designed to provide efficient storage and fast online gathering, retrieval, and update of data.
- IMS logging is more developed than CICS logging. DBRC is built into the system instead of being a messy addition.
- CICS code and IMS code are maintained separately, so there is no longer any need to regenerate CICS whenever IMS maintenance is applied.
- Failure isolation. The DBCTL interface is designed so that a failure in CICS should not cause DBCTL to fail, and a failure in DBCTL should not cause CICS to fail.
- Workload in CICS regions is reduced, so users may be able to merge CICS regions together.
- There is no longer any need to maintain MRO links that were used only for remote database access.
- DBCTL can share data in an MVS/ESA sysplex.
- Operational flexibility is a gain. CICS and DBCTL are independent of each other; that is, CICS can be running while DBCTL is not, and the converse.
- DBCTL enables you to do a number of operations online, including:
  - Online image copy.
  - Online change.
  - Online reorganization for DEDBs.
- Throughput on multiprocessors increases because the DBCTL interface resides in separate address spaces, and because DBCTL uses a separate task control block (TCB) for each application thread. In addition to higher throughput on multiprocessors, there can be more concurrent activity.

See 3.2, "DBCTL Overview" on page 20 for more detailed information. There are also some points users need to consider:

- Database security is more complicated. Users can employ RACF to limit which CICS regions can use which databases. Many security features are available to users, but each must also be administered.
- All CICS regions accessing the same DBCTL must be in the same LPAR.
- IMS-DBCTL and IMS-DC do not run under VSE.

- Older releases of CICS (Version 2 and earlier) do not support DBCTL.
- DBCTL does not support multiple copies of the same DBD or PSB. If multiple copies of the same DBDs and PSB exist, then each must be renamed. Changing the first letter of the name is the most commonly used technique. Some application products provide their own interface to achieve this.

1.4 Scope of This Book

As we indicated earlier, there will be heavy migration activity from the VSE platform to the OS/390 platform within the emerging market countries in the foreseeable future. The migration activity will coincide not only with their rapid economic development but also with the use of advanced information technology and centralized control activities.

Although VSE and OS/390 are somewhat similar, they do have enough difference to worry those who are new to the OS/390 world. We, therefore, use a basic step-by-step approach to guide new OS/390 users on how to migrate their VSE CICS local DL/I application into the new and fascinating OS/390 CICS TS and DBCTL environment. With samples we provide, readers should be able to generate their own CICS TS and IMS DBCTL environment without any difficulty.

We also understand that due to rapid economic development and limited resources available in these emerging market geography locations, migration activity may be necessary for business, but it may be something they have too little time to prepare for. Thus, in the planning and interoperability chapters (Chapter 6, “Planning Migration” on page 41 and Chapter 11, “Interoperability” on page 195), we show users how to migrate their application systems incrementally, so that migration will interfere only minimally with an already heavily loaded production environment. This is not a highly technical manual, but rather a process guide that explains to the reader, who is new to OS/390 world, what steps to take. There are some technical areas, but in general, we cover the ideas of how to proceed and in what order.

To those who already are users of early versions of MVS and want to upgrade to the latest operating system environment, Chapter 3 on IMS DBCTL and Chapter 4 on CICS TS can help you to generate the new environment more easily.

Many books have already been published on this topic, including *VSE to MVS Differences and Migration Guide, GG66-3113* and ITSO redbook *VSE to OS/390 Migration Workbook, SG24-2043*. Some of the books dealt with early versions of the MVS environment and therefore information is a bit dated. Another presents the general guidelines but lacks detailed steps and implementation samples. This book will complement the two books mentioned and present new OS/390 users with the complete picture.

Furthermore, migration is usually to a well known and technologically mature environment. This book is not a leading-edge, high-technology book but rather a collection of well known information and techniques.

1.5 Project Environment

We are using the following operating system and subsystems for the project migration target environment.

OS/390 V2R4 Product 5647-A01. However, some proof-of-concept testing jobs also run under OS/390 V1R3 (5645-001) and V2R5

IMS DBCTL IMS/ESA V6; Product 5655-158.

CICS TS V1R2; Product 5655-147. Because of actual customer migration requirements, we also run some proof-of-concept jobs under CICS/ESA V4R1 (5655-018) environment.

IMS DB Tools V2R3; Product 5685-093 and IMS Database Control Suite 5797-D15.

Language COBOL for OS/390 and VM; V2R1 Product 5648-A25. High Level Assembler for MVS and VM, V1R2; 5696-234.

Language Environment Language Environment for MVS and VM; V1R5 Product 5688-198.

The base environment is a VSE application from an IBM China customer, the Industry and Commercial Bank of China (ICBC), a large commercial bank with the following:

VSE/ESA V2R1.1; Product 5750-VSE.

CICS VSE V2R3; Product 5686-026.

Application The major online application is IBM SAFE II, the system for an advanced financial environment interactive interface, an online financial and banking application system. Others are customer in-house developed application programs based on SAFEII macros. Compilers used vary from old DOS COBOL to newer VSE COBOL II.

This project selected ICBC operation environment as the sample migration environment because ICBC is one of the largest commercial banks in China and has branches not only throughout China but overseas. This customer of IBM China is undergoing a major redesign of its EDP infrastructure and will have many migration and consolidation projects in the near future. The ICBC current application environment is a standard operating environment throughout all ICBC branches. Many other banking and financial institutions in China also have similar environments. Although it is no longer supported by IBM (a newer replacement is available), the IBM SAFE II financial application system has been heavily used not only in China, but in many other VSE environments throughout the world. The redbook from this project can assist many users who are considering an upgrade of their EDP facility to OS/390 environment.

We installed the base VSE system into one VSE guest under ITSO's Poughkeepsie Center VM system. The OS/390 environment is part of the current ITSO OS/390 sysplex. We used LPAR SC62TS for most of the testing work. Figure 4 on page 7 shows the testing configuration.

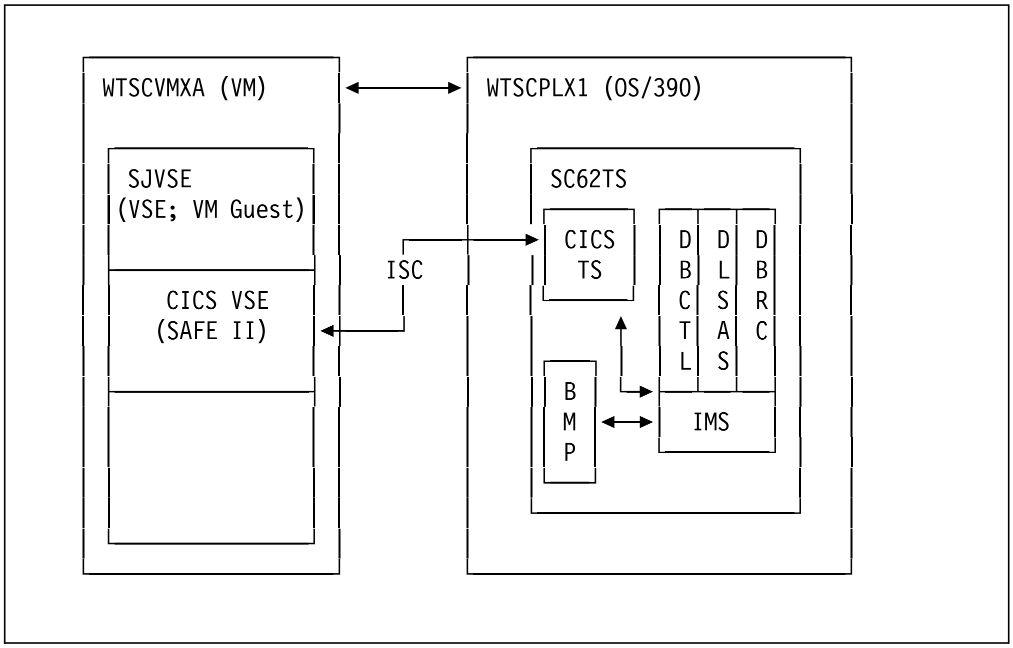


Figure 4. Project Testing Environment

1.6 Additional Projects

In addition to the basic migration of CICS/VSE with Local DL/I to CICS/ESA on OS390 with DBCTL, users may well find that several other migrations may have to be performed as part of the overall project. Many of these will result from having built user applications a long time ago on older versions of CICS and VSE; until now, users may have had no reason to use newer techniques introduced by more recent versions of software. These will be covered in more detail in later chapters. Some common examples are:

- Migrate CICS/VSE to OS/390 CICS TS or CICS/ESA V4.
- Migrate application programs from the CICS macro-level to the CICS command-level API.
- Rewrite any special exits with CICS gateway macros and CICS SPI instead of control block DSECTS.
- Migrate from DOS/COBOL or COBOL II to COBOL for OS/390
- Migrate assembler 370 programs to the High Level Assembler.
- Migrate from local security to external security.
- Migrate DL/I databases from FBA to CKD disks.
- Migrate networking environment.
- Migrate VSE JCL to OS/390 JCL.

Our book here primarily focuses on CICS/VSE with Local DL/I migrating to CICS/ESA (or CICS TS) with IMS DBCTL on OS/390. For details about other subsystem differences between VSE and OS/390; for example, POWER vs. JES, ICCF vs. ISPF, file management, and so on. Please refer to another ITSO redbook, *VSE to OS/390 Migration Workbook, SG24-2043*.

We also provide worksheets, shown in Appendix A, “Migration Planning Worksheets” on page 229, to help users with migration planning.

Chapter 2. Introduction to OS/390

A computer′s architecture consists of the functions the computer system provides. It is distinct from the physical design and, in fact, different machine designs may conform to the same computer architecture. Enterprise Systems Architecture/390 (ESA/390) is IBM′s large computer architecture. There are three operating systems based on ESA/390 architecture: VSE/ESA, VM/ESA, and MVS/ESA.

An operating system is a group of related programs that govern the computer system. The operating system controls the execution of programs and provides services they need to make use of the computer system hardware.

MVS/ESA and VSE/ESA are two very different operating systems, but they share a common instruction set. As a result, applications developed for one can often run on the other. IBM has developed and marketed the two for different markets for many years; VSE is for small systems, typically several CICS partitions with hundreds of terminals and some gigabytes of data; whereas OS/390 is for much larger systems, with thousands of terminals and thousands of gigabytes of data.

However more recently the ranges of the two operating systems have converged, so that the user can now use either VSE or MVS/ESA for all but the largest configurations. This allows users to choose which operating system to use on the basis of the facilities each offers.

2.1 Introduction to MVS/ESA

MVS/ESA is an operating system that is capable of *multiprogramming* which means executing many programs concurrently. MVS/ESA can also manage *multiprocessing*, which is simultaneous operation of two or more processors that share various system hardware devices.

Many users running many separate programs means, along with large amounts of complex hardware, that MVS/ESA users need large amounts of storage to ensure suitable system performance. The sophisticated application programs that access large data bases and program modules require the operating system to provide routines to protect privacy as well as routines for sharing the data bases and software services. The description of such complex management is beyond the scope of this book and we give only a basic introduction to the configuration of MVS/ESA.

- **Virtual Storage**. The acronym MVS stand for *multiple virtual storage* to indicate that each user has access to virtual, rather than only real, main storage. Virtual storage means that each running program can access all of the main storage that the addressing scheme allows. MVS/ESA supports a 31-bit address scheme that allows a program to address up to 2,147,483,648 bytes (2 GB) of storage locations.
- **Address Space**. A complete 2 GB range of 31-bit virtual storage addresses is known as an *address space*. MVS/ESA provides each user (batch job initiator, time-sharing option (TSO) user, or started task) with a unique address space and maintains the distinction between the code and data belonging to each address space. MVS/ESA also provides *cross memory*

services, permitting a single user to access other address spaces when necessary.

The ability of many users to share the same resources also implies the need to protect users from one another and protect the operating system itself.

- **Task management**. MVS/ESA breaks each job into separate units of work known as tasks and attempts to process each one as efficiently as possible. The tasks for one job compete with one another, and with tasks related to other jobs, for use of system resources. Responsibility for controlling the progress of tasks through the system lies with the *supervisor*, a component of the operating system. The supervisor allocates resources and maintains current information about each task so that processing can resume from the appropriate point in case of an interruption.
- **Resource management**. Multiprogramming and multiprocessing create the need to measure the activity of the system and to adjust the workload to fit changing conditions. The *system resources manager*, a component of MVS/ESA, monitors how much each active address space uses the processors, I/O devices, and real storage locations. Then it uses this information to determine whether an address space should remain in real storage or whether a new address space should be created.
- **I/O and data management**. Nearly all tasks involve some amount of data input or output. The channel subsystem manages the use of I/O devices while MVS/ESA associates the data for the task at hand with a device. MVS/ESA supports many different devices for data storage.
- **Job management**. MVS/ESA provides several ways to enter work into computer system:

  Batch processing A user enters a job through a local terminal or, by means of remote job entry (RJE), through a remote terminal, or from tape, card reader, or disk, and the system processes the job later.

  Interactive job entry The system responds to terminal users while they are actually logged-on to the system.through a remote terminal, or from tape, card-reader, or disk, and the system processes the job at a later time.

  START command MVS/ESA also permits authorized users to enter jobs by means of the START command.

- **Recovery management**. MVS/ESA includes a recovery mechanism to prevent a user error from causing the failure of the computer system, to isolate and recover from operating system errors, and to protect the system from hardware errors. It also has programs that track system activity and display the status and contents of various system resources.

MVS/ESA, in general, is a combination of program and data modules. Large groups of modules that make a particular MVS/ESA function possible are known as *system components*. Other groups of modules that provide added function dependent on MVS/ESA are known as *subsystems*. MVS/ESA also includes *subsystem interface* for communication with IBM subsystems or user-supplied subsystems.

2.2 MVS/ESA Address Space

Conceptually, an MVS/ESA address space consists of 2 GB of virtual storage available to each user. Figure 5 shows an address space for virtual storage.

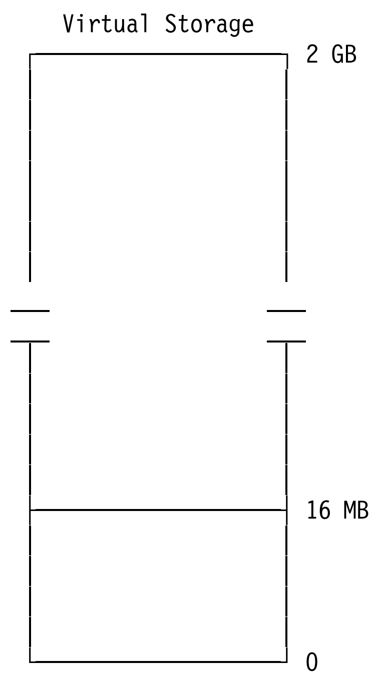


Figure 5. A Logical Representation of Virtual Storage

An MVS/ESA address space contains the system prefix save area, private areas, and common areas. Each user has an entire address space and thus has access to all three kinds of areas. MVS/ESA effectively isolates one address space from another by means of segment and page tables. Through the common area of address space, users can share programs, and data areas.

The mapping of an MVS/ESA address space is shown in Figure 6 on page 12. Except for the user region and extended user region, all other areas are used by the system program modules and data. The locations are according to the characteristics of the modules and data such as:

- They can be shared among all address spaces.
- They can be paged or must always be backed by real storage (fixed).
- They must reside below the 16-MB line.

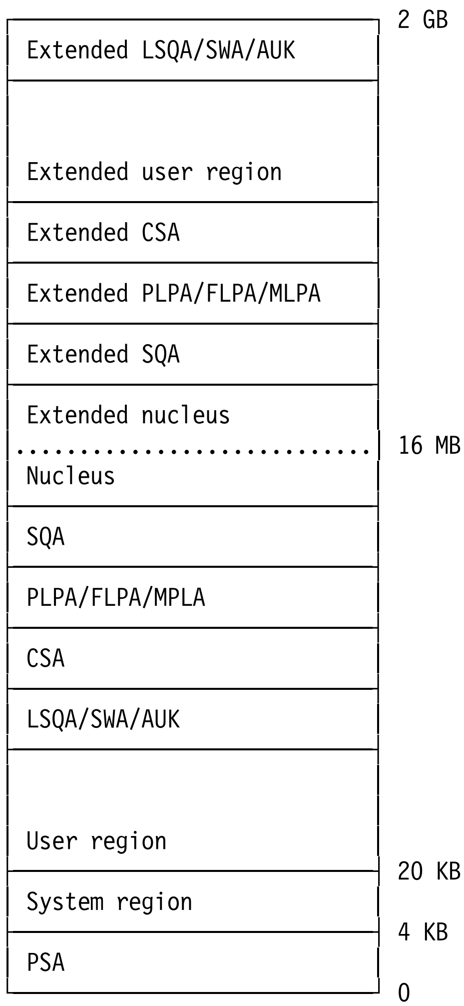


Figure 6. Virtual Storage Layout

The user region is the section of the private area in which user programs run. MVS/ESA programmers should try to use the extended user region as much as possible because it is vastly larger than user region below the 16 MB line.

2.3 MVS/ESA System Component Address Spaces

When an MVS/ESA system is initialized, many key system component address spaces are created. The first address space, called *master scheduler*, interacts with operator commands and system parameters to initiate functions. The following is a list of a few of the most important system address spaces created during system initialization: (For detailed information, the reader should refer to various MVS/ESA publications.)

1. Program call and authorization address space. It establishes cross-memory services.

2. System-trace of address space.

3. Global resource serialization.

4. Dumping services.

5. Communication task.
6. Allocation of address space.
7. System management facilities.
8. Primary job-entry subsystem address space.
9. LNKLST lookaside (LLA) address space.

MVS/ESA uses both asynchronous and synchronous communication between address spaces.

2.4 Basic OS/390

OS/390 is a set of MVS base, open, client/server, and application-enabling functions that can be ordered with one program number and delivered as one product. OS/390 consists of base elements that deliver the essential operating system functions. Most of the elements are products that have been available for some time and MVS users have probably been running some of them.

In addition to the base elements, OS/390 has optional features that have an affinity to the base. There are two types of features:

- The first type of feature is shipped with OS/390, whether ordered or not. It supports dynamic enablement. When you want to use a shipped but disabled feature, let IBM know and it will be enabled dynamically.
- The second type of feature is not capable of dynamic enablement and is not shipped unless the user specifically orders it.

2.4.1 OS/390 Base Product

The first release of OS/390 V1R1 contains the following major elements (not a complete list):

- Base Server elements:
  - MVS/ESA SP Version 5.2.2 and its associated subsystems:
    - JES2 Version 5.2.0
    - HCD
    - OpenEdition and its associates services subsystems
    - ESCON directory
    - SOMobjects for MVS
  - Bulk Data Transfer (BDT)
  - BookManager Read function
  - DFSMS
  - EREP/MVS
  - GDDM including PCLK and OS/2 link
  - ISPF and TSO/E
  - Language environment for MVS
  - SMP/E
- Communications server elements

- VTAM version with AnyNet/MVS
- TCP/IP

2.4.2 OS/390 Optional Features

The OS/390 V1R1 also has the following optional features (not a complete list):

- BookManager build function
- Bulk Data Transfer addition functions
- IBM C/C++
- GDDM-REXX
- JES3
- TCP/IP CICS and IMS sockets
- Security Server (RACF)
- RMF
- Language Environment Data Encryption Standard (DES)

The most widely used optional feature is RACF. IBM did not include this in the base product because other security products are available in the market.

2.4.3 OS/390 V2R4 Enhancement

The OS/390 V2R4 has been available since September of 1997. In addition to many enhancements to the base product of V1R1 and functions in V1R2 and V1R3, there are new features in V2R4:

- Domino Go Webserver.
- Integrated cryptographic services facility (ICSF).
- Distributed computing environment application support.
- Encina toolkit for transactional remote procedure calls.
- Book server enabling user to provide documentation on WWW.
- Hardware configuration manager

These functions have nothing to do with this project; however, interested readers should refer to IBM publications for more information.

2.5 Difference between VSE and MVS/ESA

There are many differences between VSE and OS/390, as well as a few similarities caused by their common instruction set. Users need to be aware of all of these when migrating.

This topic is not a key area to be discussed in this book, for detailed information refer to the book *VSE to OS/390 Migration Workbook, SG24-2043*.

2.5.1 VSE and MVS/ESA Similarities

Many program products can be obtained for either system, such as

- CICS
- DL/I
- VSAM
- DITTO
- VTAM
- COBOL.
- ACF/VTAM,ACF/NCP...

There are also a few architectural similarities between the two operating systems which make it relatively simple for users to convert applications from VSE to MVS/ESA, and to pass messages and data between VSE and MVS/ESA based systems. These are:

- Both use 8 bit bytes and the EBCDIC character set.
- Both use the 390 instruction set.
- Both can use 24 bit addressing.
- Both can run as guests under VM/ESA

2.5.2 VSE and MVS/ESA Differences

In spite of the similarities, there are many differences between the two operating systems which support staff will experience. The major ones are:

- **On-line access to datasets**. In VSE support staff have limited on-line access to libraries through ICCF. With MVS/ESA, staff and users have TSO with ISPF, which is a very much more powerful tool.
- **Security**. VSE has very limited security for ICCF users and none at all for batch processing. CICS security is implemented by a sign-on table, which forces CICS users to enter a password, set once by the security administrator, and defines which classes of transactions a user may access. The next version of CICS will use SAF to protect resources via external security.

  MVS/ESA is supplied with the RACF security server as standard, which users can tailor to provide whatever level of security they need. This can range from simply checking user passwords when they log on at a terminal, to the very tightest protection of all resources to the level required by high security military and financial institutions. CICS on MVS/ESA uses RACF to protect whatever resources you wish to almost any level.
- **CICS dynamic storage area size**. CICS/VSE is still constrained to limited DSA size, but CICS transaction server places most management modules and many control blocks above the 16 MB line. This provides considerable virtual storage constraint relief in the user application environment.
- **Disk file**. VSE and MVS/ESA support a number of different file organizations. Users can access VSAM unique clusters on CKD disks from both VSE and MVS/ESA, but no other file architecture is fully compatible across both operating systems. Although users are strongly advised not to share data across VSE and MVS/ESA systems, your users may well want to move data

between them. If they intend to do this, they need to be aware of the differences. Table 1 on page 16 gives a summary of these.

Table 1. VSE vs MVS/ESA Comparison

MVS/ESA	VSE
Supports CKD devices.	Supports CKD devices.
Does not support FBA devices.	Supports FBA devices.
Supports VSAM catalogs.	Supports VSAM catalog only.
Normally uses ICF catalogs.	Does not support ICF catalogs.
Cannot access VSAM-managed SAM files	Can use VSAM to manage non-VSAM SAM files.
ICF catalogs do not support VSAM data spaces.	Often define VSAM files within VSAM data spaces.
VSAM share options 1, 2, 4 do not protect against VSE VSAM access.	VSAM share options 1, 2, 4 do not protect against MVS VSAM access.

- **Job Control**. VSE uses POWER to control job submission, whereas MVS/ESA uses JES2 or JES3. There are a number of differences between them, and the way that the operating system runs jobs:
  - Different operator command format
  - Different JCL statement format
  - VSE/POWER uses DISP=K (replace job on input queue after processing) to rerun the same JCL repeatedly. MVS/ESA uses started tasks.

  See 8.8.7, "Rewrite All JCL" on page 123 for more information. An example is given in the next section.
- **JCL**. VSE JCL differs significantly from MVS/ESA JCL. The same job would look very different if coded for VSE than for MVS/ESA. An example of such a job is shown in Figure 7 and in Figure 8 on page 17 in both VSE and MVS/ESA formats:

```
* $$  JOB JNM=IMSJOB,CLASS=.....
// JOB IMSJOB accounting-info
// DLBL IMSUCT,'IMSCAT',,VSAM
// EXTENT SYS001,IMS001
// DLBL OUTDD1,'IMS.SJIMSD.DUMP',.........
// EXTENT SYS011,VOL011,...........
// ASSGN SYS011,DISK
// ASSGN SYS001,DISK,VOL=IMS001,SHR
// DLBL IMSDBASE,'IMS.SJIMSD.IMSDBASE',,VSAM,CAT=IMSUCT
// LIBDEF PHASE,SEARCH=(IMS.RESLIB)
// EXEC DFSPROG
/*
/&
* $$ EOJ
```

Figure 7. Example of Job Control Language for VSE

```
//IMSJOB JOB (acounting-info),CLASS=  .......
//STEP1    EXEC PGM=DFSPROG
//STEPLIB  DD DISP=SHR,DSN=IMS.RESLIB
//OUTDD1   DD DSN=IMS.SJIMSD.DUMP,DISP=MOD
//IMSDBASE DD DSN=IMS.SJIMSD.IMSDBASE,DISP=SHR
//
```

Figure 8. Example of Job Control Language for MVS/ESA

The differences between MVS and VSE operating environment are summarized in Table 2.

Table 2 (Page 1 of 2). VSE vs MVS/ESA Comparison

Function/Subsystem	**OS/390**	**VSE**
Instruction set	Uses 390 instruction set.	Uses 390 instruction set.
System abend code	3 HEX digits.	2 HEX digits.
Addressing mode	31-bit or 24-bit addressing.	31-bit or 24-bit addressing.
Transaction system	CICS/ESA, CICS/TS.	CICS/VSE.
Database system	Runs IMS/DC, IMS/DB, DB2.	DL/I, DB2 for VSE or SQL/DS.
Security	RACF security server.	No security product.
Interactive system	Extensive on-line access via TSO/ISPF.	Limited on-line access via ICCF.
Spool management system	JES2 or JES3.	POWER.
Communication system	ACF/VTAM, ACF/NCP, TCP/IP.	ACF/VTAM, ACF/NCP, third-party TCP/IP.
AFP printing system	PSF, OGL, PPFA	PSF, OGL, PPFA
WAN support	VTAM supports cross-domain and cross-network connections	VTAM supports cross-domain connections only.
Network management	TME 10 Netview for OS/390	Netview
Language	COBOL, PL/I, C++, REXX, LE, ...	COBOL, PL/I, C/370, REXX, LE, ...
Distributed transaction process	CICS MQ	CICS MQ
LAN support	ADSM, LANRES	ADSM, LANRES
System maintenance and installation	SMP/E	MSHP
Address spaces	up to 32,768	up to 212
Multiple processor support	Fully utilize 10 CPs	Best utilize four CPs
DASD supported	CKD disks, RVA 2 Snapshot	CKD and FBA disks.
Parallel sysplex	Yes	No
Multiple CP dispatch	Subtask level	Job (Partition) level

Table 2 (Page 2 of 2). VSE vs MVS/ESA Comparison		
Function/Subsystem	**OS/390**	**VSE**
I/O device number supported	4096	1024
HW assistant compress	VTAM, DB/2	VTAM, VSAM KSDS only
9221 integrated communication adapter	No	Yes
Dynamic reconfigure management support	Yes	No
HCD	Yes	No
CICS SGF/SSP	Yes	SOD
Hiperbatch/Hiperspace	Yes	No
DCE/POSIX	Yes	No
Domino	Yes	No

Chapter 3. Introduction to IMS/ESA DBCTL

The move from VSE CICS with local DL/I to OS/390 CICS with DBCTL combines the problems of moving:

- VSE to OS/390
- COBOL, PL1, or 390-assembler from VSE to OS/390
- VSE LE to LE for MVS and VM
- CICS on VSE to CICS/TS on OS/390
- DL/I on VSE to DL/I on OS/390
- DL/I from CICS-Local DL/I to CICS with IMS/DBCTL

The first three of these were not the main scope of this project, and we just mention them briefly in 1.6, "Additional Projects" on page 7. The best references are in the books *VSE to MVS Differences and Migration Guide, GG66-3113* and *VSE to OS/390 Migration Workbook, SG24-2043*. The three remaining topics are covered in this chapter and in Chapter 4, "Introduction to CICS Transaction Server for OS/390" on page 27.

In general, a VSE for a CICS application system probably runs with 200 to 300 terminals and a few gigabytes of data, where the MVS/ESA will have a much larger environment, often with thousands of terminals and terabytes of data. Recently, however the ranges of the two operating systems have begun to converge.

3.1 Local Data Language 1

CICS/VSE with local DL/I runs the database in the same CICS region as the application programs that access it and the on-line terminal handling it in a typical VSE CICS application. This is shown below in Figure 9.

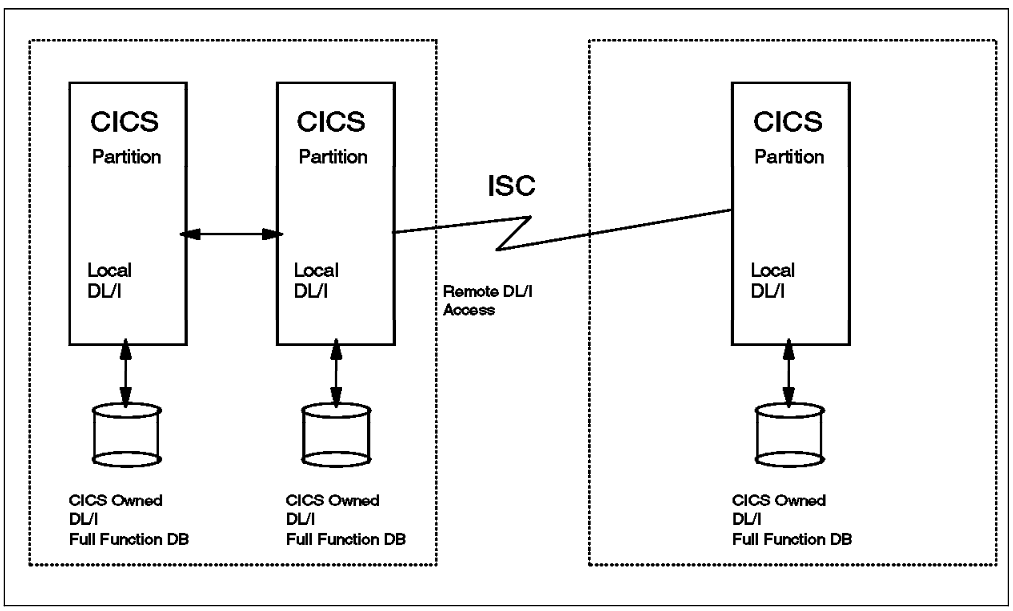


Figure 9. CICS — Local DL/I Environment

Figure 9 clearly shows that access to DL/I databases not owned by the local CICS system (remote DL/I access) is possible by means of having CICS function-shipping a DL/I request to another CICS system within the same MVS system or over an ISC link to another MVS system.

3.2 DBCTL Overview

DBCTL is an IMS facility that provides an IMS/ESA database manager (IMS/ESA DM) subsystem that can be attached to CICS, but runs in its own address spaces. DBCTL offers these benefits:

- **Release independence**. User does not need to regenerate the DL/I support in CICS on changing to a new release of CICS or IMS.
- **Access to more IMS functions for CICS users**. DBCTL gives one or more CICS systems online access to data entry databases (DEDBs) as well as full function DL/I databases.
- **Virtual storage constraint relief for CICS systems that currently contain DL/I** The DL/I code is outside the CICS address space.
- **Improved throughput on multiprocessors**. DL/I requests run under task control blocks (TCBs) separate from those used by CICS, because CICS and DBCTL reside in separate address spaces.
- **Improved logging**. DBCTL uses a separate log, the IMS log, so DL/I activity does not appear on the CICS system log. This means that all DL/I information is on a single log and can be processed using IMS logging facilities such as dual logging and database recovery control (DBRC).
- Ability to use CICS support for the extended recovery facility (XRF).
- **Improved failure isolation between CICS and IMS**. A DBCTL failure should not cause CICS system to fail.
- **Batch jobs can be run as batch message processing programs (BMPs)**. These application programs perform batch processing online, using the same DBCTL as CICS and sharing its databases, thus giving users concurrent access to IMS databases.

3.3 DBCTL Environment

Figure 10 on page 21 shows a typical DBCTL configuration. Each box represents an address space running within a single MVS/ESA system. The CICS on-line transaction runs in one region (CICS/TS); the database calls run in another (DLISAS), and extra regions run to control logging and recovery tracking (DBRC), locking (IRLM, optional), and overall control of the process (IMS Control).

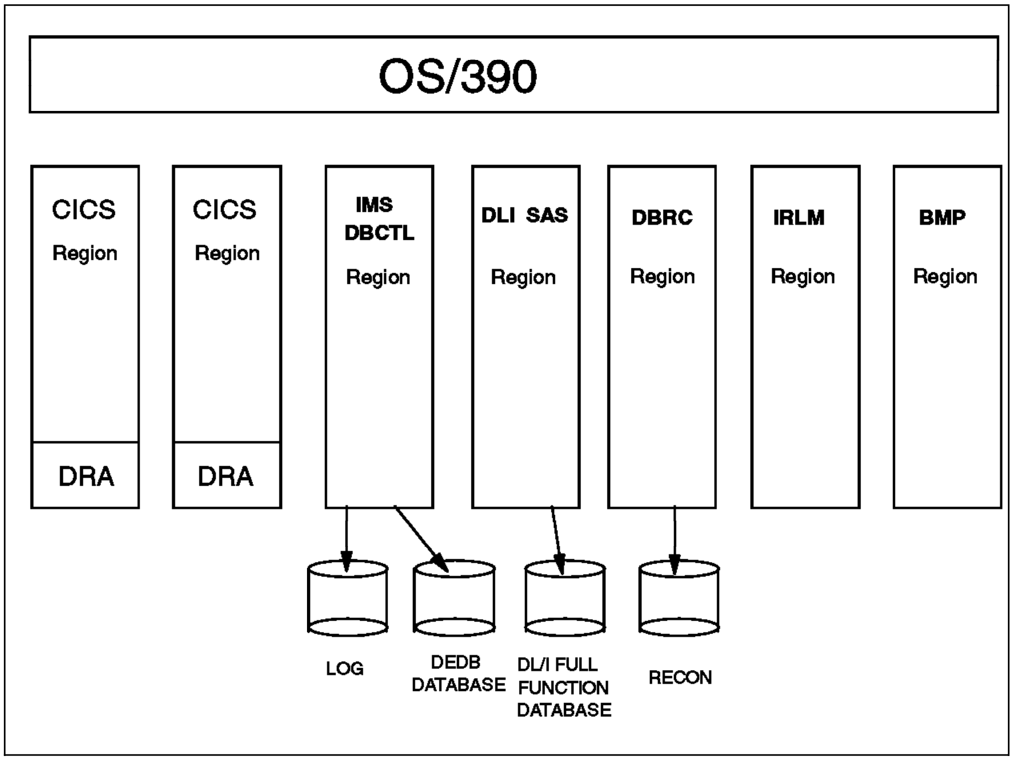


Figure 10. Typical CICS — IMS DBCTL Configuration

In spite of the different underlying configuration, a program coded with just EXEC CICS commands for local DL/I and the standard CICS API will run without change in a CICS-DBCTL environment. In practice, some program changes are needed to address other changes required in the migration, such as CICS version changes, VSE to MVS operating system changes, and the absence of old language compilers from your new OS/390 environment.

When an application is running in the CICS environment with IMS DBCTL, there are at least two more address spaces (regions) in addition to the DBCTL region itself. These are the DL/I separate address space (DLISAS) and the DBRC region. IRLM and BMP are other possible optional address spaces, which are described as follows:

DBCTL DBCTL is an environment of the IMS/ESA database manager that allows access to DL/I full function databases and the data entry database (DEDB), DBCTL can be attached to CICS, but runs in its own address space.

DLISAS DL/I full-function databases are allocated to and managed by the DL/I Separate Address Space region.

DBRC The database recovery control region is used for its IMS log control facility. DBRC manages all IMS logs, maintains control information in the RECON, and optionally controls recovery of DBs if registered, and controls the use of remote site recovery.

IRLM If databases are to be shared between two or more controlling subsystems — more than one DBCTL or between DBCTL and local CICS-DL/I — then block level data sharing must be implemented using the IMS resource lock management.

BMP Batch-message processing is used to run batch programs in the online environment. BMP can access the IMS full function database, the DEDB, and the like.

3.4 Differences between DL/I Databases on VSE and MVS

There are very few differences between the database on VSE and that on MVS. Most of the differences are caused by the different way in which VSAM is implemented in the two environments:

- Database dump and restore utilities have different names.
- Files on the same volume do not need to be catalogued in the same user catalog in MVS.
- MVS does not support FBA DASD such as 3370s.
- VSAM SHR(4) does not offer as much protection on MVS.

Most of the differences are caused by use of IMS DBCTL instead of Local DL/I on OS/390 as described in the next section.

3.5 Environmental Differences

Table 3 lists the differences when DL/I is processed in the VSE CICS environment and in the OS/390 DBCTL environment.

Table 3 (Page 1 of 2). Differences between Local DL/I and IMS DBCTL

DL/I on VSE	**DBCTL on MVS**
CICS needs to be regenerated whenever IMS maintenance is applied.	CICS does not need to be regenerated when IMS maintenance is applied.
CICS needs to have different load libraries for regions with different levels of DL/I.	CICS needs different load libraries only if CICS level changes are applied.
CICS has to close for database maintenance.	CICS and IMSDBCTL stay up during database maintenance.
DL/I databases are not available when CICS is down, except for IMS BATCH.	DL/I databases remain available for BMPs or other CICSs when CICS is down.
Database contention between a CICS transaction and a batch MRO job can often cause CICS to hang.	Database contention between a CICS transaction and a BMP stops only the BMP or the CICS transaction.
Database changes are recorded in the journal of whatever CICS system performed the update.	All database changes are recorded in the IMS logs.
CICS logs are copied to tape if set up. Operator intervention is required to avoid overwriting of CICS journal before archiving.	Archiving of IMS logs to tape (or disk) is automatic — no operator intervention or CICS or IMS hanging.
PDIR and DDIR definitions in assembled CICS table.	PDIR and DDIR definitions in IMS MODBLKS or FULL generation.
New PSB and DBD definitions require a restart of CICS.	New PSB and DBD definitions can be installed while CICS and IMS are running.

Table 3 (Page 2 of 2). Differences between Local DL/I and IMS DBCTL	
DL/I on VSE	**DBCTL on MVS**
DBD and PSB names must be unique within a CICS region.	DBD and PSB names must be unique across all CICS systems using the same DBCTL.
Single operator interface available through CEMT.	Operator interfaces CEMT, CDBM, and SYSLOG all required.
Database security controlled by CICS access security.	Database security controlled by IMS AGN security in RACF.
Database command security controlled by CICS access security.	Database command security controlled by RACF access to console commands (RACF) and also by CICS access security to CDBM command.
DBRC is optional.	DBRC is mandatory for work initiated from CICS and for BMPs, and optional for independent batch jobs.
Only database types (S)HISAM, HDAM and HIDAM supported.	Database types (S)HISAM, HDAM and HIDAM supported, also Fast Path, DEDBs, and all other types allowed in IMS.
CICS with DL/I is a single region.	CICS with DBCTL consists of four or five regions.

3.6 Migration Consideration

IBM offers a clear migration path for those who use early versions of MVS CICS (that support local DL/I) and want to migrate to the latest CICS TS in OS/390. Unfortunately, there are no published recommendations or suggestions for VSE migration procedures. In addition, none of the later versions of either CICS TS or IMS/ESA systems support local DL/I, which means that no intermediate migration steps are available (migrate to MVS CICS with local DL/I then to the DBCTL environment). The entire migration processing is a one-step procedure and is heavily dependent on personal skill and experience. The goal of this project is to guide the entire migration processing. This section makes some recommendations for migration to DBCTL. Chapter 6, “Planning Migration” on page 41 has a detailed description of the entire migration planning aspect.

Since no intermediate migration steps are available, users will probably want to migrate to DBCTL in stages, perhaps as follows:

1. Set up a test system. If the users already have a test system for testing new applications, consider using it for testing migration to DBCTL.
2. If the user does not want to begin with a test system, begin by setting up a trial production system, perhaps one already used for testing existing production applications.
3. Set up a production DBCTL.

The user should then:

- Generate DBCTL, DLISAS, and DBRC.
- Decide which applications to migrate.
- Convert CICS shared-database programs to BMPs.

- Convert any programs that use DFHFC TYPE=DLI macros to issue DL/I commands or calls instead.
- Convert VSE CICS region to OS/390 CICS TS.
- Tune CICS-DBCTL.
- Convert batch jobs to BMPs, which must issue checkpoints.
- When migrating CICS shared-database programs to BMPs, define program specification blocks (PSBs) in DBCTL security generation.
- Consider DEDBs for new applications.

3.7 Planning New DBCTL Environment

The following are some environmental considerations to be taken into account:

- If the user is running multiple CICS regions, each with its own copy of local DL/I, we recommend migrating all local DL/I systems to a single DBCTL.
- If a user is running the same applications that schedule the same PSBs on each of the CICS regions, but access different instances of the same databases, migrating to a single DBCTL means a need for a separate DBD and separate PSBs for each instance of a database. However, user applications could continue to schedule the same PSBs and use the CICS global user exit, called *XDLIPRE*, to change the PSB name, the SYSID that the application program has scheduled at execution time, or both. See *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683* for programming information on using this exit.
- In a remote DL/I environment, in which multiple CICS application-only regions (AORs) function ship DL/I requests to a DL/I resource-owning CICS region in the same MVS image, replace the DL/I resource-owning region with DBCTL. However, you cannot replace the DL/I resource-owning region with a DBCTL subsystem in a different MVS image.
- CICS TS systems running in a separate MVS image from DBCTL must function-ship their DL/I requests to a CICS TS located in the same MVS image as DBCTL.
- It is possible to share IMS databases across multiple DBCTLs in a single- or multi-MVS environment.

Furthermore, Chapter 11, “Interoperability” on page 195 gives methodology suggestions for those who want to migrate large and complex applications.

3.8 Setting Up DBCTL Environment

Consider following points when setting up test and production systems.

3.8.1 Number of DBCTL Subsystems

Users should plan carefully to determine the number of DBCTLs required in a single-MVS environment. DBCTL needs MVS common storage area (CSA); therefore, the amount of available CSA is the most important factor in determining the number of DBCTLs within a single MVS image.

We recommend having only one production DBCTL in a single-MVS environment. Normally, this should be large enough to serve all CICS TS for OS/390 systems within one MVS image. For multiple CICS systems running in several MVS images, count the number of DL/I threads needed. If the sum of these threads, plus the number of expected active BMPs, is less than 255, you should need only one DBCTL.

We also recommend using one log for each DBCTL. Logging can become more complex the more DBCTLs users have. Balance the need for multiple DBCTLs against the logging procedures users need.

3.8.2 ACBLIB Considerations

DBCTL needs a minimum of two application control block libraries (ACBLIBs), if you are using the IMS online-change function. Online change is not new in IMS, but it certainly is a new concept for VSE CICS users. Users need to become familiar with the concept and operation.

Chapter 4. Introduction to CICS Transaction Server for OS/390

The Customer Information Control System (CICS), is a general purpose and large scale database and data communication system that runs on almost all IBM and IBM-compatible machines. Also, there are versions of CICS that run on non-IBM hardware. CICS Transaction Server for OS/390 is the version of CICS that runs on the IBM OS/390 platform. It can either be in a sysplex configuration or run outside of one. There are many publications, ITSO redbooks, product manuals, and articles in various magazines about this subject, ranging from introductory level to in-depth technical descriptions and analysis. Here, we present a basic introduction to those who are new to the OS/390 world, limited to the scope of this project. We do not cover other functions CICS TS has. Interested readers should obtain more detailed information from the reference sources mentioned.

4.1 Overview

CICS Transaction Server for OS/390 is an S/390 client/server package that provides CICS server, CICS client, and CICS management functions. It meets the latest information technology requirements:

- Secure integration of business with the new world of electronic commerce and network computing for easy integration of business systems for both intranet and Internet.
- Flexible point of entry into network computing with a scalable and reliable growth path.
- Comprehensive client support from IBM and non-IBM workstations.
- Support for all models of client/server computing.
- Exploitation of S/390 Parallel Sysplex.

The CICS Transaction Server for OS/390 base release (Version 1 Release 1) includes the latest network computing technology, such as recovery management, system management, and enhanced client/server and internet support. In addition, CICS TS improves the use of the VTAM generic resource function and uses the coupling facility technology for handling log stream and recovery. The detailed information can be found in various production publications as well as ITSO redbooks. We outline its major features here:

- Support S/390 Parallel Sysplex which includes additional data sharing facility:
  - VSAM record level sharing (RLS) among multiple CICS systems.
  - CICS temporary storage data sharing among CICS systems across multiple MVS images.
- Centralized sysplex-wide log and journal management, sharing CICS transaction log streams
- System management including single enterprise-wide point of control, monitoring, and workload balancing
- Enhanced interface to the World-Wide Web
- CICS recovery manager, which provides coordinated recovery actions for an entire CICS configuration

- VTAM generic resource support improvements
- Resource definition online (RDO) for transient data
- New EXEC CICS CREATE command for dynamic creation of resource definitions in a CICS TS region

4.2 CICS Transaction Server for OS/390 V1R2

With the needs of a sysplex environment and multiple CICS regions running across several MVS images, there is a requirement for improved journal control, recovery mechanisms, data sharing, and resource availability. The CICS TS 1.2 provides the following new features that we summarize here:

- Support for single MVS image systems using the DASD-only logging function of OS/390, provided in OS/390 V2R4. It supports both a single image in a sysplex configuration without a coupling facility (nonparallel sysplex) and stand-alone OS/390 systems (single-system sysplex). It offers an upgrade path for all MVS customers and VSE CICS users who want to migrate to OS/390.
- New interface that allows 3270-based CICS transactions to run unchanged without a 3270 terminal
- Enhanced interface to the World Wide Web (WWW) adds support for 3270-based transactions
- Support for the CICS gateway for Java
- Enhanced REXX for CICS
- New system management facility for defining and installing CICS resources across multiple CICS
- New RDO for DB2 resource definitions
- Added client/server capability, with support for client partner LU6.2 applications across TCP/IP network

ITSO redbook *CICS Transaction Server for OS/390 V1R2 Implementation Guide, SG24-2234* has a very good description of the CICS Transaction Server for OS/390 V1R2.

4.3 CICS Transaction Server Log Manager

One of the most noticeable CICS difference, from the VSE user point of view is that CICS journal files no longer exist in the CICS Transaction Server for OS/390 world. They are replaced by the new CICS TS log manager.

The new CICS log manager, which replaces the journal control management function of earlier releases, works with the MVS system logger to provide a focal point for all system log, forward recovery log, and user journal output within a single MVS sysplex. Using services provided by the MVS system logger, the CICS log manager supports:

- The CICS system log, which is also used for dynamic transaction backout
- Forward recovery logs, automatic journals, and user journals (general logs).

It also maps CICS journals onto MVS log streams, which can be used, for example, to merge all the forward recovery logs for a given VSAM data set from many CICS regions on several MVS images online. The MVS system logger also enables faster CICS restart, dual logging, and journal archiving.

The CICS log manager, with the MVS system logger, improves management of CICS system log, automatic journals and user journals. It simplifies recovery procedures and operators, so the danger of operational error is minimized. The CICS log manager also:

- Avoids log wraparound.
- Automatically deletes obsolete log data of completed units of work (UOWs).

The CICS log manager can retrieve data directly as well as sequentially, and most system log data is immediately available, thus making CICS restart processing faster.

4.4 DASD-Only Logging

For most of the VSE users who are migrating to the OS/390 environment, sysplex is probably not their first step. We suggest a single OS/390 system configuration (one MVS image) because it allows users to become familiar with the OS/390 environment and acquire necessary technical skills for the next step. DASD-only logging may be used. The function is available in CICS TS 1.2 with OS/390 V2R4. A DASD-only log stream (new in OS/390 V2R4) connects to a single MVS system. Multiple CICS regions on the same MVS image can use the same DASD-only log stream.

When an application writes a log block to a DASD-only log stream, the MVS system logger writes the block first to the local storage buffers for the system and then automatically duplexes it to a DASD staging data set associated with the log stream. The DASD staging data set is the primary medium for DASD-only logging. When the staging data set runs out of space, it off-loads to the secondary storage medium.

4.5 Differences between CICS/VSE and CICS/TS

CICS/VSE is at Release 2.3. CICS/TS is at Release 1.2. There were a lot of changes to CICS between these two levels. Many of the changes introduced into CICS on MVS to free storage constraints were not appropriate to VSE with its smaller available region sizes. Others, like the use of an external security manager, were not appropriate to VSE because no such product is installed on most sites.

4.5.1 Summary of Differences

Table 4 is a summary of the main differences between CICS/VSE and CICS TS.

Table 4 (Page 1 of 2). Differences between CICS/VSE 2.3 and CICS/TS 1.2

CICS/VSE 2.3	CICS/TS 1.2
DL/I databases available only through local DL/I.	DL/I databases available only through IMS DBCTL.
Macro-level programming allowed.	Macro-level programming not allowed.
Does not support RDO for FCT and DCT entries.	Supports RDO for FCT and DCT entries.

Table 4 (Page 2 of 2). Differences between CICS/VSE 2.3 and CICS/TS 1.2	
CICS/VSE 2.3	**CICS/TS 1.2**
Supports macro definitions for PPT and PCT.	Does not support macro definitions for PPT and PCT.
Macro definitions for VTAM TCT allowed.	Macro definitions for VTAM TCT for migration to RDO only.
No dynamic allocation of data sets.	Application files can be dynamically allocated in FCT.
No dynamic allocation of data sets.	Application files can be dynamically reallocated by CEMT.
Enter commands at start-up through CSMT on a sequential terminal.	Enter commands at start-up through a PLT program with parameters passed by SIT.
No autoinstall of PPT definitions.	Allows autoinstall of PPT definitions.
No temporary install of resource definitions.	Temporary install of resource definitions by CECI CREATE / DISCARD.
Security provided by assembled SNT.	Security provided by RACF or other external security product.
User password changed by user administrator only.	User must change own password at regular intervals.
Supports file sharing using VM minidisk sharing, or DISP=SHR.	Supports file and temporary storage sharing over Sysplex, or DISP=SHR.
No dynamic routing of transactions.	Allows dynamic routing of transactions.
No support for generic VTAM application ID.	Can use a generic VTAM application ID for Sysplex.

4.5.2 SIT Differences

The system initialization table (SIT) parameters vary from release to release, and there are fewer of them in the VSE version of CICS. This is one of the most important tables for the system environment. For reference, we summarize, in Table 5, parameters used in both VSE CICS and CICS/ESA V4R1 but that no longer exist in the CICS TS environment. The reason we include CICS/ESA V4R1 is that the software is still available and is Year 2000 compliant. It works well on a nonsysplex single-MVS-image environment. CICS/ESA V4R1 is the only system that supports local DL/I in an MVS environment. VSE users may want to consider migrating to this version from VSE to gain technical skill, then migrating again to the OS/390 DBCTL environment. Table 6 on page 32 shows new SIT parameters on CICS TS only.

Table 5 (Page 1 of 3). SIT Parameters in CICS VSE and CICS/ESA V4 Only			
Parameter	**CICS ESA 4.1**	**CICS VSE 2.3**	**Remark**
ADDUMP	X		Produce dump when abend with code ASRB
ALEXIT	X		Exit program on transaction initiation
ALT	X		Application load table
AMXT	X		Maximum number of tasks allowed for dispatching

Parameter	CICS ESA 4.1	CICS VSE 2.3	Remark
AUTINST	X		Maximum number of devices queued for autoinstall
BFP	X		The built-in functions program
CMXT	X		The maximum number of tasks
COBOL2	X		VS COBOL II support
DATFORM	X	X	Date format
DBP	X	X	Dynamic transaction back-out program
DBUFSZ	X	X	Dynamic log buffer
DDIR		X	List of data management blocks
DLDBRC		X	Presence of DBRC
DLI	X	X	DL/I databases to be accessed
DLIOER	X		DL/I I/O error
DLIOLIM		X	The number of DL/I I/O errors allowed
DLIRLM		X	Presence of IRLM
DLLPA		X	Location of IMS modules
DLMON		X	DL/I database monitoring
DLTHRED		X	The number of threads for local DL/I
DLXCPVR		X	Page fix for ISAM or OSAM buffer
DMBPL		X	DMB pool size
DTB	X		Location of dynamic log records
ENQPL		X	ENQ control block
EXEC	X		Command level support
EXITS	X		User-exit interface
EXTSEC	X		External security manager support
FERS	X		Support of facility error-recognition system
ICVS		X	RDO parameter
JCT	X	X	JCT suffix
MONITOR	X		Monitor class
PCDUMP	X		Produce dump when abend wirh code ASRA
PCT	X		PCT suffix
PGSIZE	X		CICS virtual page size
PISCHD		X	Program isolation scheduling
PLI	X		PL/I application support
PPT	X		PPT to be loaded
PSBPL		X	PSB pool size
SCS	X		Storage cushion
SIMODS	X		Initialization overlays
SVD	X		Storage violation dump
TRACE	X		Number of trace entries

Table 5 (Page 2 of 3). SIT Parameters in CICS VSE and CICS/ESA V4 Only

Table 5 (Page 3 of 3). SIT Parameters in CICS VSE and CICS/ESA V4 Only			
Parameter	**CICS ESA 4.1**	**CICS VSE 2.3**	**Remark**
TSMGSET	X	X	Number of entries for dynamic storage
XRFTODI	X		The takeover delay interval
XRFTRSZ	X		The size of the CAVM trace table
XSWITCH	X		The programmable terminal switching unit
ZCP	X		The suffix of ZCB, ZCZ, and ZCX parameters

We also summarize the new SIT parameters available only for CICS Transaction Server for OS/390 in Table 6.

Table 6. New CICS TS 1.2 SIT Parameters

Parameter	**Remark**
CSDINTEG	Level of read integrity for the CSD when accessed in RLS mode
CSDRLS	Access the CSD in RLS mode.
DB2CONN	Start the DB2 connection automatically.
DBCTLCON	Start the DBCTL connection automatically.
ECDSASZE	ECDSA size
ERDSASZE	ERDSA size
ESDSASZE	ESDSA size
EUDSASZE	EUDSA size
FTIMEOUT	Time-out interval for files, RLS mode
MQCONN	Start the MQ connection automatically.
NEWSIT	Load new SIT on warm start.
OFFSITE	Restart is taking place at a remote site.
PRVMOD	Name of modules not to be used from LPA
QUIESTIM	Time-out value for data set quiesce requests
RDSASZE	RDSA size
RLS	VSAM RLS mode
RLSTOLSR	Strings and buffer number calculation include RLS files
SDSASZE	SDSA size
SDTRAN	Shut down transaction name
SPCTRxx	Level of trace on CICS xx domain
STNTRxx	Standard trace on CICS xx domain
TDINTRA	CICS is to initialize with empty intrapartition transient data queues.
UDSASZE	UDSA size
UOWNETQL	Specifies a qualifier for the NETUOWID for units of work initiated in the local CICS region.
WEB	Start the CICS WEB connection automatically
WEBDELAY	WEB delay period
XDB2	CICS to perform DB2ENTRY security checking

Chapter 5. Application Environment

We are going to focus on the application languages used on this project only. As mentioned in 1.5, "Project Environment" on page 6, the customer banking application SAFE II was developed in IBM assembler language and many of the back-end batch report programs used IBM's COBOL language. IBM offers the following application development tools for both VSE/ESA and OS/390 platforms (Table 7).

Table 7. IBM Application Environment

Language	OS/390	VSE/ESA
COBOL	COBOL for OS/390 and VM; V2.1. Product 5648-A25	COBOL for VSE/ESA; V1.1. Product 5686-058
Assembler	High Level Assembler for MVS and VM; V1.2. Product 5696-234	High Level Assembler for VSE; V1.1. Product 5696-234
PL/I	PL/I for MVS and VM; V1.1.1. Product 5688-235	PL/I for VSE; V1.1. Product 5686-069
Language Environment	Language Environment for MVS and VM; V1.5. Product 5688-198	Language Environment for VSE; V1.4. Product 5686-094

Table 7 implies that when coding a high-level assembler application program, it will be fully compatible between VSE and OS/390 environment. Although there are different product numbers for COBOL for OS/390 and VSE, application programs coded according to the new COBOL standard are generally compatible between OS/390 and VSE environment from an application development point of view. The difference is in the ability to support 24- or 31-bit addressing for a reentrant module when placed at shared virtual area (SVA) of VSE or common area in an OS/390 environment.

5.1 Assembler

In general, the object module produced by High Level Assembler is portable across all the supported operating systems. Therefore, an assembler program can be assembled on any of the supported operating systems and run on any of the supported operating systems. For example, an assembler program can be assembled under MVS and run under VSE (however, it must relink with the VSE linkage editor).

Also, users must be careful of system-supplied macros. Some macros may not be fully compatible across all operating systems.

5.2 DOS COBOL

During our planning and inventory taking session, we found that many of the batch programs were developed under early versions of VSE DOS COBOL and, as expected, many of the SAFE II on-line CICS application programs used CICS macro-level application program interface (API). We therefore include a few words about DOS COBOL and migration aid here. A completed migration methodology is shown in Chapter 6, "Planning Migration" on page 41

The current COBOL for VSE language is based on the COBOL 85 standard and includes all features of its predecessor, IBM VS COBOL II. It also supports the IBM SAA COBOL level 2 common programming interface. We strongly recommend that users migrate all their application programs to this version of the compiler first, then consider migrating to the OS/390 environment.

5.3 Migrating from DOS COBOL to COBOL for VSE

Following are some rules which should be observed when planning and implementing a migration to COBOL for VSE (or to COBOL for MVS and VM also).

- Use MIGR in DOS/VS COBOL to identify incompatibilities between DOS/VS COBOL and COBOL for VSE.
- Do not overstress sharing capabilities of COBOL for VSE; that is, be very selective when loading phases into the SVA. An increase in the SVA may create virtual storage constraint problems.
- Carefully select compiler options that are appropriate for the user environment, paying particular attention to defaults set by COBOL for VSE.
- Differentiate between the potential functions of COBOL for VSE and current OS/390 implementation.

In general, the differences and incompatibilities between DOS COBOL and COBOL for VSE are minimal. The extent of change depends heavily on the programs and the environment. Users should not forget that many of the inconsistencies concern rare functions and language elements. The most common corrections are to:

- Change the ASSIGN-clause to conform to COBOL for VSE format.
- Remove base locator for linkage (BLL) cells.
- Remove CURRENT-DATE and CURRENT-TIME and replace them with DATE-IN-PROGRAM and TIME-OF-DAY in batch programs, with EIBDATE and EIBTIME in online programs.
- Replace EXHIBIT and EXAMINE with DISPLAY and INSPECT.
- Avoid new COBOL reserved words in application programs.
- Delete all Report Writer statements.

5.4 The Advantages of Migrating to COBOL for VSE

The immediate advantages of migrating to COBOL for VSE from DOS COBOL are programmer productivity (new effective language elements, structured programming), use of RETURN-CODES, and so on. Long-term advantages include portability between operating systems, and virtual storage relief with extended addressing capability. The advantages are summarized as follows:

- **Portability between different operating systems**. Programs compiled under COBOL for VSE are portable between the MVS, VM and VSE operating systems, and between CICS/VSE and CICS/ESA transaction processing environments. Programs can be run in either 24-bit or 31-bit addressing mode and exploit the extended architecture of the environments.
- **The SAA and ANSI 1985 standard support**. COBOL for VSE delivers the SAA COBOL interface for System/370 and System/390 environments. The ANSI

1985 COBOL standard provides efficient language standards. Users can write nested programs and use inline PERFORM and EVALUATE statements to construct applications in a top-down manner using structured programming and modular design.

- **Programmer productivity increase**. The language provides the opportunity to write structured programs and develop applications modularly:
  - The EVALUATE statement in COBOL for VSE is useful to replace complex IF-THEN-ELSE structure with many imbedded IF-sentences.
  - Inline PERFORM statements enable users to write PERFORM statements directly inline between the PERFORM and END-PERFORM statements. This reduces transfers within programs, and makes their logic easier to understand.
  - The source input in VS COBOL II can be written in upper case, lower case, or both upper and lower case.
  - COBOL for VSE offers a special register RETURN-CODE to pass the return code between separately compiled programs.
- **Expanded FILE STATUS clause**. The expanded clause in VSAM in COBOL for VSE enables application programs to obtain file status information as well as VSAM-return code.

5.5 Conversion Aid Programs

IBM provides a conversion aid program to assist users converting from early versions of COBOL to the current ANSI COBOL 85 standard, which is COBOL for VSE. See F.1, “COBOL Conversion Aid for VSE” on page 281 for more information about this product.

The IBM Hursley technical support center in the U.K. also offers a migration aid to convert CICS/VSE macro-level application programs to command-level API. See F.2, “CICS/VSE Application Migration Aid” on page 282, for detailed information.

5.6 Differences between COBOL for VSE and for MVS

The differences between COBOL for VSE and MVS/ESA are even less than those between DOS COBOL and COBOL for VSE, but some small changes are necessary. The following sections describe these changes, which involve only minor rewrites and recompilation.

5.6.1 CURRENT-DATE

VSE will return this field as mm/dd/yy or dd/mm/yy depending on the VSE option you selected with the

```
// DATE
```

JCL statement in a job, or the STDOPT DATE=... statement in the background partition startup. MVS always returns the date as mm/dd/yy. If the user is retrieving the date as dd/mm/yy, the application program must be rewritten. However the user would probably do this anyway, to replace COBOL calls to DATE with calls to the LE date and time services to make the program Year 2000 compliant.

5.6.2 Online COBOL Programs

Here, we cover the CICS on-line processing only, and make no attempt to discuss other on-line processing environments. VSE CICS COBOL programs written in strict accordance with the CICS API should need only to be recompiled with OS/390 COBOL for MVS and VM and linked with OS/390 binder.

5.6.3 Batch COBOL Programs

These entail more changes because users will have been able to code a much greater variety of things. The common ones are these:

- Referencing a printer I/O area before opening the file or after closing it.
- Embedding a STOP RUN within a SORT procedure.
- Using numeric values on a level 88 statement PIC X field without quotes does not work. For example

```
77   PP-VALUE-FLAG  PIC XXX.
  88 GOOD-RETURN  VALUE 000
  88 BAD-RETURN   VALUE 999 0C4 B37
```

  works in VSE but not in MVS. It must be recoded thus:

```
77   PP-VALUE-FLAG  PIC XXX.
  88 GOOD-RETURN  VALUE '000'
  88 BAD-RETURN   VALUE '999' '024' '837'
```

- 01 REDEFINES are not allowed in FILE SECTION, because these are implicit anyway within the same FD.
- SPECIAL NAMES clause. Users must change the special names shown in Table 8.

Table 8. VSE vs OS/390 COBOL SPECIAL-NAMES Entries

VSE	OS/390
SYSLST	SYSPRINT
SYSPCH	SYSPUNCH
SYSIPT	SYSIN

  If users make extensive use of these names in programs, they may need to code statements in SPECIAL-NAMES of the form

```
SYSPRINT IS SYSLST.
```

- FILE-CONTROL. Remove RESERVE xx ALTERNATE AREAS clauses and recode SELECT - ASSIGN clauses as follows:
  - Remove SYSnnn.
  - Remove device. (MVS allows specification in the JCL.)
  - Change UR class specification to UT
  - Change BDAM organization field, A to D, and U to W.

An example would be:

```
VSE     SELECT MAP305 ASSIGN SYS011-UR-1403-S.
MVS     SELECT MAP305 ASSIGN UT-S-MAP305.
```

- BDAM files using absolute track addressing must be recoded to use relative track addressing. Also the track position part of the key is coded as PIC S9(8). OS/390 codes it as PIC S9(5).
- ISAM files. Users should convert any ISAM files to VSAM KSDSs and rewrite programs accordingly.
- INPUT-OUTPUT CONTROL. Remove SAME AREA, APPLY EXTENDED SEARCH, APPLY WRITE-VERIFY and all ISAM-specific clauses.
- INPUT-OUTPUT CONTROL, MULTIPLE-FILE clause. MVS treats this clause as comments, so make sure you code the file position on the LABEL clause of the DD statement for the file.
- DATA DIVISION; FILE SECTION. Remove BLOCK CONTAINS clause and always use LABEL RECORDS STANDARD.
- PROCEDURE DIVISION. Convert user labels to standard labels. Also, COBOL ERROR-STATUS field is 136 bytes instead of the 8 in VSE. Users must rewrite any error routines that access this field as it contains the full system message.
- STOP RUN and GOBACK on VSE cause an end-of-job (EOJ) whereas in MVS they may simply return to an intermediate layer, for example, PL1 invoking program.
- ACCEPT without FROM is no longer supported.
- Relational operators within brackets are not allowed. For example, IF A < (C OR D) is not allowed. Recode as IF (A < C) OR (A < D).
- Users cannot specify BLANK WHEN ZERO and also specify an * as a zero suppression symbol for the same data item.
- Users can no longer link-edit an assembler program onto the end of a COBOL program and transfer control by omission of a termination statement. Users must explicitly CALL the assembler program.
- MOVE CORRESPONDING can now have only one receiving data item. For example, MOVE CORRESPONDING GROUP-A TO GROUP-B GROUP-C must be recoded as two statements, MOVE CORRESPONDING GROUP-A TO GROUP-B and MOVE CORRESPONDING GROUP-A TO GROUP-C.
- If users move oversize data into a full-word binary field, the result may be different in IBM COBOL. Compiler option TRUNC(OPT) is almost the same as the old option NOTRUNC.
- Lower case letters now pass the ALPHABETIC class test. The keyword ALPHABET is now required.
- COBOL programs calling assembler programs with the CALL statements cannot pass PARAGRAPH names, only parameter fields.
- COBOL for VSE does not permit a COPY statement in a 01 level statement in the DATA division. Instead, users must use a COPY REPLACING statement immediately after a *01 DATAXX.* statement.

- VSAM file status value is different from SAM file status.

  There is still a 2-byte status code for SAM files,

```
77 SAM-STATUS        PIC XX.
```

  but the return code for VSAM files is 8 bytes.

```
01 RETURN-STATUS.
   05 FS-CODE            PIC X(2).
   05 VSAM-CODE.
      10 VSAM-R15-RETURN  PIC 9(2)  COMP.
      10 VSAM-FUNCTION    PIC 9(1)  COMP.
      10 VSAM-FEEDBACK    PIC 9(3)  COMP.
```

- Use of UPSI byte. This is covered in 8.8.6, "Rewrite Programs Accessing the VSE UPSI Byte" on page 122.
- CURRENT-DATE. The CURRENT-DATE is called an *intrinsic function*. It returns a 21-character alphanumeric value that represents the calendar date, time of day, and time differential from Greenwich Mean Time.

5.7 Difference between VSE PL1 and PL1 on MVS

No application programs were written in PL1 on this project, so we omit any detailed discussion on PL1. The following subsections state some obvious differences and the user should refer to the product publications for more detail.

5.7.1 Compiler Options

There are three main differences:

- The CATALOG option is not supported in OS/390. Use a

  `//SYSLIN DD DSN=....`

  statement instead.
- LIMSCONV is not supported.
- LINK is not supported. Use conditional JCL instead.

There are other differences, but all are available only on OS/390.

5.7.2 Execution Options

5.7.2.1 Online Programs

- ISASIZE: User should take care to resize this for every on-line program.
- SPIE and STAE options: User must set these options to on. Otherwise, CICS may not handle program abends, and may abend itself.

5.7.2.2 Batch Programs

No changes needed.

5.7.3 Environment Attributes

The following options are not supported under OS/390 and will cause compilation errors:

- FUNCTION
- ASSOCIATE

Many other options are not supported, but they are simply ignored by the compiler.

5.7.4 PL1 Optimizer

The DUMP options D and ND are not supported under OS/390.

5.7.5 PL1 Overlay Structures

If users employ the overlay structure, they must remove CALL PLIOVLY statements and replace them with linkage editor options PARM=OVLY, INSERT, and OVERLAY. Since there is much more region storage under MVS, users normally need not build overlays.

5.8 Difference between VSE LE and LE/MVS

VSE LE and LE/MVS are designed to give a common run-time environment for programs written in any supported language (COBOL, PL1, Assembler, or FORTRAN). Differences between VSE and OS/390 are minimal.

Chapter 6. Planning Migration

Converting an application from one operating platform to another is not like a normal application development project. Consider these questions:

- Which application should be migrated first?
- What is the makeup of the application programs?
  - How many are there?
  - Do they all use the same language?
  - How good is the current documentation?
  - Do all programs need migrating?
- Are there other application systems that share data with the system targeted for migration?
- If we copy the shared data to the new environment, how will updates be synchronized?
- Can the application system be split so that it can be migrated in stages?
- How good is the data model of the current system?
- What is the quality of the data in the current system?
- What skills are needed for the migration project?
- How much time is available for the project?
- Should other departments be involved in the migration?

These are just some of the consideration that must be fully addressed. Careful study and planning is required to ensure success. We present here a series of methodologies for consideration. We try to make each step relatively small. The basic required information can then be found and a decision made in a short time. This allows decision makers to allocate fewer resources to investigating feasibility. The approach method is this:

1. **Identify the requirement**. The first step is to identify the requirement. This entails identifying what benefits are expected from this new replacement. From an overview of the current system, planners must identify which options will fulfill most requirements to produce a cost effective result.
2. **Test the feasibility**. The next step is to test the feasibility of the output of Step 1. This is done by putting a plan in place and testing it with pilot replacement software to see if the assumptions made are correct.
3. **Carry out the main conversion**. The major replacement work takes place in this step, implementing the plan worked out in Steps 1 and 2.
4. **Implement the application**. Finally, use the new replacement system for production.

6.1 Planning Aspects

The objectives of the migration are to maximize the functions and capabilities of the new operating system. Intelligent analysis and careful planning are needed. The following items describe a suggested working profile:

- **Survey**. Any document systems are candidates along with their programming languages and data sources. Such a survey sets the boundaries for the subsequent project.
- **Portfolio analysis**. The goal of this analysis is to get an idea of the state of the systems, the size of the task, the effects of use of data by multiple systems, and any special difficulty you must accommodate.
- **Personnel**. One issue is the impact of a new project effort on available resources. For this reason, we suggest that an outside organization, IBM or another consulting organization, be used. Methodologies are suggested that do not require the involvement of the original operational personnel. Resources must be available to the new team that works on the migration, but the majority of the team need not be the original people.

  There are benefits to considering a consulting group for this effort, since the skills of mapping the current application system into the new operating environment may not be required after the migration is completed. Also, using outside consulting services means that the team can act relatively independently and need not consume resources used in the normal course of business.
- **Staging**. It is unlikely to be reasonable or practical to replace the entire system in one step. This guide is written assuming a phased effort in which part of the application processing is replaced by the corresponding part of the new operating environment.
- **Performance**. An important question during any migration process is the impact of the new system on performance. There are a number of aspects to performance (for example, CPU time, wall clock time, number of physical I/O′s, and memory utilization). All aspects are important, but their relative importance often relates to the particular environment and the resources that are most critical at the time of conversion.

  As in all performance issues, the actual results experienced by a customer will depend on many variables, and may differ from customer to customer.

6.2 Planning Considerations

Some level of planning must be done for the overall migration project that encompasses education and environment requirements, and the initial analysis of the current operation process.

For the first migration effort, users should have a plan that addresses the major tasks and subtasks for analysis and implementation within the overall framework. Subsequent migration projects can have a similar plan but reflecting actual experience. The project plan can be finalized as soon as the workload analysis is completed. At a minimum, the plan should include measurable checkpoints, completion criteria, resource requirements, team structure, cost estimate, and scheduled completion date. Many project planning tools are available; the IBM Application System (AS) is one of many on the market.

The methodology used for estimating cost should take the following factors into account:

- Inventory of data and related programs.
- Estimated complexity.
- Estimated cost per unit of work.
- Cost of added hardware, both permanent and temporary.
- Staff requirement.
- Education.

The project should be tracked as closely as possible to provide feedback for measuring the accuracy of the initial project schedules and cost estimates. At the completion of the project, the actual cost can be used to validate and adjust the estimating techniques used. Also, be sure to consider learning curves and orientation, as these bring a one-time cost incurred during the first conversion project.

6.3 Education

Education is a key factor in any conversion project. Education and understanding subject matter mean confidence on the team. A realistic assessment of the skills of the team will give the project manager a base for education planning. Not all education requirements are necessarily met by formal classroom programs. Considerable skill transfer takes place in informal on-the-job training, meetings, judicious use of consultants, and self-study.

6.4 Hardware and Software Considerations

In general, no special hardware is required for migrating an application system from VSE/ESA to OS/390. However, at least one additional CPU is needed to house the new OS/390 operating system. Where the current CPU is large enough to divide into multiple partitions, one PR/SM LPAR is needed to install the new operating system.

In general, the CPU requirements are not significant in the beginning. The final testing and production cutover, with both current operation and the new replacement running similar workloads, is likely to impose the peak CPU load. This peak load may be the sum for current operation in both environments.

Portions of files must be replicated for the purpose of test and production cutover. Most testing can be done with files of minimal size. At the point of production cutover, full-size files or databases must be used for verification of peak DASD usage. Planning to replace in phases rather than all at once reduces the DASD requirement, especially for those who are already constrained.

6.5 System Testing

A comprehensive test plan is important. The plan needs to have measurable tests, acceptance criteria, test exit criteria, schedule dependencies, and task relationships agreed to. The test plan also needs to cover fallback and recovery synchronization options.

Testing should be done in an environment as stable as possible. Therefore, it is essential to minimize version changes to the operating system, the compilers, or any other programs that may cause the results to differ from those of the original system.

Review any change to the environment during the processing to see if new test conditions are needed. All design or operating changes should have associated documentation. This documentation is used to assist with test definition.

Testing may be done in the following sequence:

1. Define and identify tests.
2. Document the test plan and criteria to be used for acceptance.
3. Create tests according to the plan.
4. Create test files or databases. These should be a small subset of the current production files. The small amount of data will minimize the cost and reduce testing time.
5. Create procedures that will subsequently be used for production operation.
6. Commence testing using test files or databases.

6.6 Verify the Results

The output of each test should be examined for equivalence with the output of current operation. VSAM databases can be unloaded into sequential data sets and use standard ISPF SUPERC command for comparison. Identify and reconcile the differences. Where the differences are not acceptable, change, and retest.

Before cutting over to production, do a trial run. If time constraints prevent an entire production operation, make sure you use a large enough representative sample for operation testing.

6.7 Implementation

Once all verification is completed, the replacement system must be put into production mode. Use the same procedures as in the earlier stages of testing to ensure the operation is run in the way you tested it.

6.8 Project Worksheets

Users should plan the migration project carefully and document the plan. Users need not only to ensure that each step has completed successfully before they start the next one, they must also be aware of what they would need to do to fall back at each stage, and what point you can fall back to.

We provide a set of work sheets in Appendix A, “Migration Planning Worksheets” on page 229 for reference. The worksheets list all possible aspects of a migration project in a orderly way and should be useful for most applications. We recommend using the worksheets as the starting point for migration, especially for those who are new to OS/390.

6.9 Project Migration

The remainder of this chapter describes a particular migration that we had done. However, the best way to plan your migration for your particular installation could well be different, and may not need to include all these steps. Equally, you may face considerations that did not apply here.

We begin by addressing those steps that can be done separately in advance, then look at which ones have to be done as part of the main migration. We also look at the dependencies between the tasks.

First of all we itemize the migration tasks. Details are to be found in Chapter 8, “Application Migration” on page 81. For each task, we consider these four aspects:

- Implementation dependencies
- Impact of implementation
- Approximate time involved
- Back-out considerations

Here is a summary list of the steps the user needs to perform. 6.10, “Detailed Task Explanation” on page 46 gives details.

- Premigration tasks:
  1. Install OS/390 with CICS/TS, IMS/DBCTL.
  2. Migrate CICS tables program control table (PCT) and program property table (PPT) to RDO on VSE
- Main migration tasks:
  1. Copy all current DL/I DBDs and PSBs to OS/390.
  2. Recode DBDs and assemble into IMS DBDLIB.
  3. Recode PSBs and assemble into IMS PSBLIB.
  4. If you are an MVS user, copy CICS DDIR and PDIR into IMSGEN source.
  5. If you are an MVS user, recode CICS PDIR for remote entries only.
  6. Run IMS MODBLKS generation (plus security generation and copy jobs).
  7. Run ACBGEN on OS/390.
  8. Code and assemble IMS dynamic allocation modules for databases.
  9. Migrate DL/I databases from VSE to OS/390.
  10. Register databases with DBRC.
  11. Copy the CICS CSD from CICS/VSE to CICS/TS
  12. Copy all current CICS tables to OS/390.
  13. Migrate CICS TCT and FCT to RDO on OS/390.

14. Recode the CICS SRT.
15. Assemble and link any CICS tables.
16. Make chance to accommodate the fact that JCT, ALT, and NLT are no longer used.
17. Migrate from CICS journals to the OS/390 MVS system logger.
18. Migrate from CICS internal security to external security.
19. Rewrite CICS exits.
20. Migrate from DOS/COBOL or COBOL II to COBOL for OS/390.
21. Migrate assembler application programs to ASMA90 assembler.
22. Rewrite applications for Year 2000.
23. Rewrite programs using UPSI byte.
24. Migrate CICS programs from MACRO-level to Command-level.
25. Migrate VSE DL/I batch to IMS non-BMP batch.

- Postmigration tasks:
  1. Install full external security.
  2. Recode IMS batch jobs as IMS BMPs.
- Optional migration tasks
  1. Migrate CICS DCT table to RDO.
  2. Implement autoinstall for terminal and program definitions.

6.10 Detailed Task Explanation

Here are the detailed descriptions of the tasks listed at 6.9, "Project Migration" on page 45. The actual work is described in Chapter 7, "System Environment Customization" on page 57, and Chapter 8, "Application Migration" on page 81.

6.10.1 Premigration Tasks

The term *premigration* means the point at which planning steps have already taken place and the migration decisions have been made. Hardware and software have been ordered and delivered (if not already installed). At this point, the tasks are these:

1. Install OS/390 with CICS TS, IMS DBCTL and other related utilities.
   - **Implementation dependence**. This task should be started as soon as is possible. It will not affect the application in the slightest. All steps in the main migration stream depend on completion of this implementation. We assume new installation here; an existing MVS installation could require far fewer installation steps. It depends on the current environment.
   - **Impact of implementation**. The impact is minimal if it is a new OS/390 image. On the other hand, if implementing CICS TS and IMS/ESA on an existing OS/390 image, schedule an IPL as part of the implementation. If there are earlier releases of CICS or IMS on the same OS/390 image, then carefully study the impact and consider a test of the new SVCs for the new CICS and IMS. Also, test for any LINKLIST and LPALIB changes. If the user wants to use existing SVC numbers for CICS and IMS, since

SVCs are downward compatible, the new SVCs will work with existing older versions of those products. This book has assumed that the user is installing into a new OS/390 image.

- **Approximate time-scale**. One to two months to install and test, if user is installing from scratch. IMS and CICS will each take about one to two weeks, depending on the skill level of the team, and on whether or not OS/390 is already installed.
- **Back-out considerations**. None.

2. Migrate CICS tables PCT and PPT to RDO on VSE

   - **Implementation dependence**. This can be done at any time before the main migration starts. This must be done before the CSD is migrated to MVS. The user should test the migration, running CICS on VSE with the migrated tables before signing this task off.
   - **Impact of implementation**. The migration to the VSE CSD can be done without impact to CICS, even in the VSE production environment. Testing it out in VSE will require a ***cold*** start of CICS.
   - **Approximate time-scale**. One hour to a few days, depending on the quantity of work and the team′s skill level. Testing will take longer, as CICS will need to be restarted.
   - **Back-out considerations**. This can be backed out simply by adding PCT=xx and PPT=xx again and cold-starting CICS. However, make sure that the previous VSE tables library is always available.

6.10.2 Main Migration Tasks

Because there are rather a long list of tasks to be done on main migration, we are dividing them into three subsections to describe functions. The main migration tasks are, actually, rather mechanical and not complicated. However, attention is needed since many steps must be performed. Also, some steps must be run in sequence while others could be run in parallel.

6.10.2.1 Migration Tasks for IMS DBCTL

The next few items are IMS/ESA related tasks.

1. Copy all current DL/I DBDs and PSBs to OS/390.

   - **Implementation dependence**. This can be done at any time once OS/390 is installed.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Half an hour to half a day depending on the team′s skill level.
   - **Back-out considerations**. This can rerun as many times as desired.

2. Recode DBDs and assemble into IMS DBDLIB.

   - **Implementation dependence**. This can be started as soon as OS/390 and IMS are installed. It must be completed before any database migration can be done.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Several days depending on the number of DBDs (about half an hour per DBD).

- **Back-out considerations**. Any error will cause a rerun of the DBDGENs and ACBGENs that follow, and a required ONLINE change. The ONLINE command sequence is:

```
<IMSD>MOD PREPARE ACBLIB.
<IMSD>MOD COMMIT.
```

  where IMSD is our project region ID.

3. Recode PSBs and assemble into IMS PSBLIB.
   - **Implementation dependence**. Needs the DBDs to be assembled first. ACBGEN and testing of migrated database depend on this.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Half an hour to one hour per PSB.
   - **Back-out considerations**. Rerun of the ACBGEN and on-line change are required.
4. Copy CICS DDIR and PDIR into IMSGEN source.
   - **Implementation dependence**. Needs only OS/390 available to update the source, but requires IMS/ESA installed to assemble it.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. One to two hours, depending on the team′s skill level.
   - **Back-out considerations**. If these are implemented as a copy book, then backing out consists of removal of the copy book followed by a MODBLKS gen to remove it from the generated library.
5. Run IMS MODBLKS generation (plus security generation and copy jobs).
   - **Implementation dependence**. Wait until the full installation of IMS and OS/390 has been completed. You must run these jobs before you can test the migrated database with CICS.
   - **Impact of implementation**. This is implemented by issuing the following, using these on-line change commands:

```
     <IMSD>MOD PREPARE MODBLKS.
then <IMSD>MOD COMMIT.
```

     while IMS is running.
   - **Approximate time-scale**. One to three hours, depending on the team′s skill level.
   - **Back-out considerations**. Reissuing the above on-line change commands, with IMS running, will restore the previous environment. Record the online and staging library set.
6. Run ACBGEN on OS/390.
   - **Implementation dependence**. Users can rerun this at will. We recommend that users regenerate only those ACBs that affect the new modification rather than regenerating the whole library. Otherwise, implementation must wait until IMS is not processing work. This needs the PSBs to be generated before it can be run.

- **Impact of implementation**. Implement these online commands while IMS is up:

```
             <IMSD>MOD PREPARE ACBLIB.
followed by <IMSD>MOD COMMIT.
```

- **Approximate time-scale**. Half an hour in total.
- **Back-out considerations**. Back this out by reissuing the MODIFY commands that implemented it.

7. Code and assemble IMS dynamic allocation modules for databases.
   - **Implementation dependence**. User can rerun this any time at will. Ensure that the database is offline when changing its dynamic allocation.
   - **Impact of implementation**. You can implement this online while IMS is up. However, we recommend taking the database offline with a /DBR command while changing its allocation.
   - **Approximate time-scale**. Half an hour.
   - **Back-out considerations**. /DBR of the database and reinstall the old module in USERLIB.
8. Migrate DL/I databases from VSE to OS/390.
   - **Implementation dependence**. The team should do this twice. Once to test, with OS/390 and IMS implemented, and the DBDGEN already run, and once as the final migration, before which the user must have completed all the premigration and main migration steps.
   - **Impact of implementation**. The user must stop CICS on VSE to unload its databases. If databases are already defined to IMS on OS/390, the user will either have to stop DBCTL, or issue a /DBR commands against the databases.
   - **Approximate time-scale**. Allow a full shift during a quiet time to do this.
   - **Back-out considerations**. The test migration can be repeated. The production migration must be done correctly, since once the migrated database has been updated, it cannot be backed out.
9. Register databases with DBRC.
   - **Implementation dependence**. This can be done at any time once OS/390 and IMS are installed. Databases will not run in DBCTL until the databases have been registered.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Half an hour to two hours depending on the team′s skill level and work load.
   - **Back-out considerations**. Run a job to issue DELETE commands in place of the INIT commands.

6.10.2.2 Migration Tasks for CICS Transaction Server for OS/390

The following tasks are related to the CICS Transaction Server for OS/390:

1. Copy the CICS CSD from CICS/VSE to CICS TS.
   - **Implementation dependence**. We recommend that the user migrate the PPT and PCT to the CICS VSE CSD before migrating the CSD to OS/390. It will save a lot of manual work in the later steps.

- **Impact of implementation**. None if the user is using a new CSD on MVS.
- **Approximate time-scale**. Half an hour to half day, depending on the team′s skill level. Allow more time for transferring tapes and for the unforeseen inherent in running run-once migrations using tapes.
- **Back-out considerations**. The job can be rerun as many times as required.

2. Copy over all current CICS tables to OS/390.
   - **Implementation dependence**. This can be done at any time once OS/390 is installed.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Half an hour to half a day, depending on the team′s skill level.
   - **Back-out considerations**. This can rerun as many times as desired.
3. Migrate CICS TCT and FCT to RDO on OS/390.
   - **Implementation dependence**. This can be done at any time once the CSD has been migrated to OS/390.
   - **Impact of implementation**. The CICS system will need to be restarted with a cold start to implement this.
   - **Approximate time-scale**. Half an day to to two days, depending on the team′s skill level and knowledge of application system for a complete set of tables.
   - **Back-out considerations**. Always keep the copy of the load modules (suffix MG) so that the jobs can be rerun.
4. Recode the CICS SRT.
   - **Implementation dependence**. This can be done at any time once the CICS tables have been copied to OS/390. Program rewriting may depend on this step, but it is likely that no further steps depend on it.
   - **Impact of implementation**. This needs careful testing. A warm start of CICS will implement it. A badly coded SRT could result in CICS abnormally ending, or continuing and causing data corruption when it should have abended.
   - **Approximate time-scale**. Allow one to two days to implement and test fully.
   - **Back-out considerations**. The SRT can be removed by coding SRT=NO in SIT overrides and restarting CICS at any time.
5. Recode CICS PDIR for remote entries only.
   - **Implementation dependence**. This is needed only if the new CICS/TS region will continue to access some PSBs remotely in other CICS regions, where the PSBs still run as local DL/1 databases. Needs restart of CICS to implement.
   - **Impact of implementation**. Both regions need to monitor closely.
   - **Approximate time-scale**. Half an hour to an hour, depending on skill level.
   - **Back-out considerations**. This will need a restart of CICS to back out.
6. Assemble and link other CICS tables, if present.

- **Implementation dependence**. This cannot be done until CICS/TS and IMS have been installed on OS/390. It is required before the migrated databases can be tested.
- **Impact of implementation**. A cold start of CICS is needed to implement the tables.
- **Approximate time-scale**. An hour to half a day, depending on skill level.
- **Back-out considerations**. Needs cold start CICS to back this out because DCT entries not found in the tables are rebuilt from the global catalog.

7. Migrate VSE CICS journals to the OS/390 MVS system logger.
    - **Implementation dependence**. This is required before the databases are live if it requires any recovery of transient data queues or uses recoverable temporary storage. Otherwise this is not required. Database logging is done by IMS.
    - **Impact of implementation**. None.
    - **Approximate time-scale**. Up to half a day to implement, plus any time needed for initial program load (IPL).
    - **Back-out considerations**. Logging is easily backed out by reimplementing the initial dummy logging definitions. Restart of CICS is required.
8. Implement external security to replace CICS internal security.
    - **Implementation dependence**. This should be begun as soon as OS/390 is installed. The project leader should take some time to identify what protections, if any, are needed and how they can properly be provided. The resources can all be defined, and the security activated when required by changing CICS start-up parameters. RACF can be used in warning mode for a while so that you can understand what access it needs to give authority for.
    - **Impact of implementation**. CICS will need a restart to switch security on. After that, resources can be defined as required, and activated by the RACF command SETR RESET.
    - **Approximate time-scale**. Planning can take several days. Defining resources will take half an hour to half a day, depending on skill level.
    - **Back-out considerations**. Resources can easily be deleted. Also security can be switched off again by changing CICS start-up parameters and restarting CICS.
9. Rewrite CICS exits.
    - **Implementation dependence**. This task requires OS/390 and CICS/TS (and maybe IMS/ESA) to be installed, in order to run any testing, although the source editing can be started at any time. Rewriting must be complete before the exits can be tested.
    - **Impact of implementation**. Probably none, but it may depend on the program. In general, the exit program needs to run from an APF library. If the load library is not in the APF list, extra time may be needed to get someone with the correct authority to arrange it.
    - **Approximate time-scale**. If complete rewriting is required, it can take many days, or even months, to convert the exits.

- **Back-out considerations**. Keep old version of source and load modules, so that the rewrite can be backed out freely. Don′t put the program into LPA or LINKLST until it has been tested thoroughly.

6.10.2.3 Migration Tasks for Application

These are application-related tasks:

1. Migrate from DOS/COBOL or COBOL II to COBOL for OS/390.

   - **Implementation dependence**. This task requires OS/390, IMS/ESA and CICS/TS to be installed, in order to run any testing, although the source editing can be started at any time. Migration must be complete before running the application against the production database.

     It is preferable to migrate all DOS COBOL or COBOL II to COBOL for VSE before migrating to the OS/390 environment. Refer to Chapter 5, “Application Environment” on page 33 for a detailed discussion of application environment.

   - **Impact of implementation**. Probably none, but it may depend on the application.

   - **Approximate time-scale**. For most cases, half an hour per program should be enough, but programs may occasionally take much longer.

   - **Back-out considerations**. Keep old versions of source and load modules, so that back-out can be performed freely.

2. Migrate assembler application programs to the ASMA90 assembler.

   - **Implementation dependence**. This task requires OS/390, IMS/ESA and CICS/TS to be installed, in order to run any testing, although the source editing can be started at any time. Migration must be completed before an application can be tested against the production database.

   - **Impact of implementation**. Probably none, but it may depend on the application. If the program needs to run from an APF library, and the load library is not already in the APF list, then the user will have to get someone with the correct authority to arrange this. This is also true if programs need to load into the MLPA or any other system-protected area.

   - **Approximate time-scale**. If complete rewriting is required, the team may have to spend many days converting each program.

   - **Back-out considerations**. Keep old versions of source and load modules, so that the user can back out freely. Don′t put the program into LPA or LINKLST until it has been tested thoroughly.

3. Rewrite applications for Year 2000 compliance.

   - **Implementation dependence**. This task requires OS/390, IMS/ESA and CICS/TS to be installed, in order to run any testing, although the source editing can be started at any time.

   - **Impact of implementation**. This may have implications both application-wide and in database design. If the users need to restructure the database, as to change a date field in one of the root segments, then they must make the database off-line to IMS by using /DBR commands, and run the appropriate restructure utilities. If the users plan to write a conversion program as part of the migration project, then they must allow enough time to test it as well.

- **Approximate time-scale**. This is impossible to estimate. It depends completely on the nature of the changes users must make to the application.

- **Back-out considerations**. These will again depend on the nature of the change. Before restructuring a database, the user should make an image copy of the database. Once you start to update the new database, you can go back to the point of restructure, but any work performed since would have to be redone from scratch. It cannot be recovered using IMS utilities.

4. Rewrite programs using UPSI bytes.

   - **Implementation dependence**. This task requires OS/390, IMS/ESA, and CICS/TS to be installed, in order to run any testing, although the source editing can be started at any time. UPSI bytes still function, although they are not supported by current COBOL for OS/390, so we strongly recommend rewriting application programs as soon as possible.

   - **Impact of implementation**. Probably none, but it may depend on the application.

   - **Approximate time-scale**. For most cases, one to two hours per program should be enough for a simple change, but programs may occasionally take much longer.

   - **Back-out considerations**. Keep old versions of source and load modules ready, so that you can back out freely to VSE.

5. Migrate CICS programs from Macro-level to Command-level programs.

   - **Implementation dependence**. This task requires OS/390, IMS/ESA and CICS/TS to be installed, in order to run any testing, although the source editing can be started at any time. Migration must be complete before the application can be tested against the production database.

   - **Impact of implementation**. Probably none, but it may depend on the application. If the program needs to be run from an APF library, and the load library is not already in the APF list, the user will have to get someone with the correct authority to arrange this.

   - **Approximate time-scale**. If complete rewriting is required, users can spend many days, even months, converting each program.

   - **Back-out considerations**. Keep old version of source and load modules, so that the migration can be backed out freely. Don′t put the program into LPA or LINKLST until it has been tested thoroughly.

6.10.3 Postmigration Tasks

After the main migration tasks have been done and fully tested, the following tasks could be processed in parallel with implementation work.

1. Install full external security.

   - **Implementation dependence**. Implementation of this depends on having installed external security above.

   - **Impact of implementation**. The team can do this while CICS is running simply by defining the resources to RACF and issuing the SETR command. Users are advised to plan carefully before implementing this, and to do so gradually. CICS RDO definitions can be changed as required to protect individual resources.

- **Approximate time-scale**. Do it over a 3 to 6-month period so that the impact of each change in security policy can be understood before proceeding to the next.
- **Back-out considerations**. Delete the resources from RACF, and switch CICS definitions on-line.

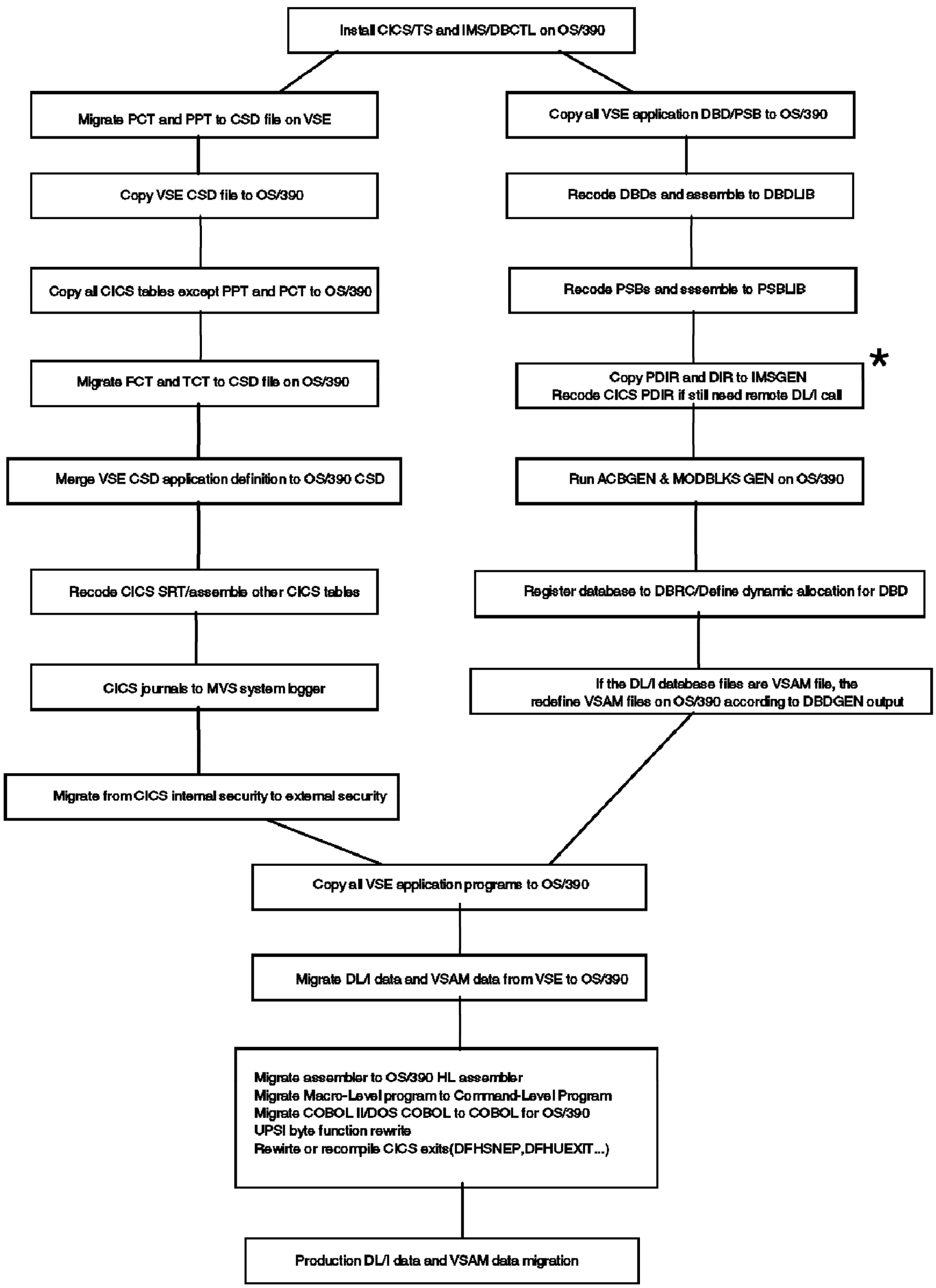
2. Convert batch DL/I jobs to BMPs.
   - **Implementation dependence**. This task requires OS/390 and IMS/ESA to be installed, in order to run any testing, although the source editing can be started at any time. Jobs can be changed one at a time.
   - **Impact of implementation**. None.
   - **Approximate time-scale**. Conversion will probably take a couple of days per program, depending on program complexity. Occasionally, programs may take much longer
   - **Back-out considerations**. It should be possible to revert back to a batch job if a load library is kept.
3. Implement autoinstall for terminal and program definitions.
   - **Implementation dependence**. If the user does not already have Autoinstall implemented on VSE for terminals, the conversion should wait until the CICS TCT and CSD have been migrated to OS/390. However, if users already have full Autoinstall for terminals on CICS/VSE, they can implement it immediately with the migrated CSD. Otherwise, they can migrate to Autoinstall all at once, or taking a group of devices at a time.
   - **Impact of implementation**. It requires a cold start of CICS without the terminal definitions in the start-up list.
   - **Approximate time-scale**. Writing and testing the terminal autoinstall program, and defining the model terminals can take up to a month. Autoinstall for programs can be implemented much faster, as the defaults supplied by IBM are generally sufficient, so that only a cold start of CICS with parameter changes is required.
   - **Back-out considerations**. Reinstall the terminal or program definitions on-line for those terminals or programs for which users want to use predefined definitions.

6.11 Third-Party Products

If any other products are used in conjunction with the current production CICS/DLI VSE system, then planners have to contact the vendors of those products to find out about migrating or replacing them.

6.12 Dependency Diagram for Migration Tasks

Figure 11 on page 55 attempts to show what tasks are dependent on which other ones, following the processing stream 6.10, “Detailed Task Explanation” on page 46.



* If you are migrating from local DL/I to IMS DBCTL on MVS environment, this step is required

Convert DL/I batch job to IMS BMP batch / Migrate DCT to CSD / Implement autoinstall for terminal and program

Figure 11. Dependency Diagram for Migration Tasks

Chapter 7. System Environment Customization

This chapter shows an example of what the CICS TS and IMS DBCTL combined system could look like initially. It is into this that the user would then migrate applications.

This chapter does not attempt to describe all the installation process for CICS and IMS, which can be found in the various manuals. The following are some of the references:

- *IMS/ESA, Installation Volume I, GC26-8736*.
- *IMS/ESA, Installation Volume II, GC26-8737*.
- *CICS Transaction Server for OS/390 Installation Guide, GC33-1681*.
- *CICS Transaction Server for OS/390 CICS IMS Database Control Guide, SC33-1700*.
- *CICS Transaction Server for OS/390 V1R2 Implementation Guide, SG24-2234*.

The last book on the list is an ITSO redbook that contains a clear description of CICS TS V1.2 implementation.

This chapter describes the installed system, so that the user can see what all the components should be for the example used in this book.

7.1 System Data Sets

System data sets include those files that stored the subsystem software and are primarily used by system programmers for software maintenance or upgrade. No daily operation functions are required for those data set. System data sets also include those files that store execution load library or common parameters stored as data sets used by application systems.

The allocation of all these system data sets taking account of their space requirement, data set placement recommendations, and so on is part of the subsystem installation processing and is not repeated here. Refer to the volumes listed as needed.

7.1.1 SMP/E Data Sets

These are used for installation and maintenance of the CICS and IMS software. As they have no other use, they can be archived to tape if an installation does not intend to operate its future own maintenance. A list of these can be found in Figure 124 on page 240 and Figure 125 on page 241 of Appendix B, "Resources Needed by Initial Installer" on page 239.

7.1.2 Run-Time and Routine Administration Data Sets

The data sets shown on Figure 12 on page 58 and Figure 13 on page 59 are needed to run day-to-day operation of CICS TS and IMS DBCTL subsystems. Some of the files are required for the usage of on-line processing, some are for off-line utilities, and others are for administering application definitions, and the like.

These data sets are installation dependent and each site has its own naming convention. The listings show what ITSO implemented and are given for reference purposes only.

```
CICS.SJCICSD.DBDC.JCL
CICS.SJCICSD.DFHAUXT
CICS.SJCICSD.DFHBUXT
CICS.SJCICSD.DFHCMACD
CICS.SJCICSD.DFHCSD
CICS.SJCICSD.DFHDMPA
CICS.SJCICSD.DFHDMPB
CICS.SJCICSD.DFHGCD
CICS.SJCICSD.DFHINTRA
CICS.SJCICSD.DFHLCD
CICS.SJCICSD.DFHTEMP
CICS.SJCICSD.MACLIB
CICS.SJCICSD.PARMLIB
CICS.SJCICSD.SDFHAUTH
CICS.SJCICSD.SDFHLOAD
CICS.SJCICSD.TABLOAD
CICS.SJCICSD.TABSRC
```

Figure 12. CICS General Usage System Data Sets

7.1.3 Setting Up Catalog Aliases

We set up the following catalog aliases for this project.

IMS.*

CICS.*

Where '*' means generic data set with above high-level qualifier. The catalog alias is mainly to facilitate system maintenance and easy administration. It depends heavily on installation naming conversion. The statements needed to define these are shown in B.2.2, “IDCAMS Statements” on page 242.

```
IMS.SJIMSD.ACBLIB
IMS.SJIMSD.ACBLIBA
IMS.SJIMSD.ACBLIBB
IMS.SJIMSD.DALSRC
IMS.SJIMSD.DBDC.JCL
IMS.SJIMSD.DBDLIB
IMS.SJIMSD.DBRC.SYSMDUMP
IMS.SJIMSD.DLISAS.SYSMDUMP
IMS.SJIMSD.EXEC
IMS.SJIMSD.EXITS.RESLIB
IMS.SJIMSD.JCLLIB
IMS.SJIMSD.JOBS
IMS.SJIMSD.MACLIB
IMS.SJIMSD.MATRIX
IMS.SJIMSD.MATRIXA
IMS.SJIMSD.MATRIXB
IMS.SJIMSD.MODBLKS
IMS.SJIMSD.MODBLKSA
IMS.SJIMSD.MODBLKSB
IMS.SJIMSD.MODSTAT
IMS.SJIMSD.OBJDSET
IMS.SJIMSD.OLP00
IMS.SJIMSD.OLP01
IMS.SJIMSD.OLP02
IMS.SJIMSD.OLP03
IMS.SJIMSD.OLP04
IMS.SJIMSD.OLP05
IMS.SJIMSD.OLS00
IMS.SJIMSD.OLS01
IMS.SJIMSD.OLS02
IMS.SJIMSD.OLS03
IMS.SJIMSD.OLS04
IMS.SJIMSD.OLS05
IMS.SJIMSD.OPTIONS
IMS.SJIMSD.PGMLIB
IMS.SJIMSD.PROCLIB
IMS.SJIMSD.PSBLIB
IMS.SJIMSD.RDS
IMS.SJIMSD.RECON1
IMS.SJIMSD.RECON2
IMS.SJIMSD.RECON3
IMS.SJIMSD.RESLIB
IMS.SJIMSD.SLDSP.D*.T*.V*   ) Archives of OLDSn. Only create
IMS.SJIMSD.SLDSS.D*.T*.V*   ) SLDSS if requires second copy.
IMS.SJIMSD.STAGE1.SOURCE
IMS.SJIMSD.SYSMDUMP
IMS.SJIMSD.USERLIB
IMS.SJIMSD.WADS0
IMS.SJIMSD.WADS1
IMS.SJIMSD.WADS2
```

Figure 13. IMS General Usage System Data Sets

7.1.4 Setting Up Initial RACF Profiles

We set up the following RACF profiles with universal update access to enable our project members to access these data sets:

IMS.**

CICS.**

IMS.SJIMSD.**

CICS.SJCICSD.**

where ′**′ means that RACF is protecting all data sets have a high-level qualifier immediately prior the ′**′. For example, IMS.RESLIB, IMS.SJIMSC.GENLIB, and so on, all are protected. Again, readers are reminded that access privilege is installation dependent and differs from site to site. Follow company IT security policy on this. The statements needed to define these are shown in B.2.3, "RACF Definitions" on page 242.

7.2 Authorize Libraries for CICS and IMS

Many IMS DBCTL and CICS TS system modules run on system area of the OS/390 address space. A feature called APF informs OS/390 that such modules exist and resides on the listed library. Without APF, MVS will not allow you to load a module into such an area for processing. The following libraries must be APF authorized:

- IMS.SJIMSD.RESLIB
- IMS.SJIMSD.USERLIB

  Library IMS.SJIMSD.USERLIB is an optional library. It does not need APF if is not concatenated with system library *RESLIB*.

- IMS.SJIMSD.MODBLKSA
- IMS.SJIMSD.MODBLKSB
- IMS.SJIMSD.MATRIXA
- IMS.SJIMSD.MATRIXB
- CICS.SJCICSD.SDFHAUTH

See Appendix B, "Resources Needed by Initial Installer" on page 239 for the statements to do this.

7.3 Constructing CICS TS Application Region JCL

A CICS TS region can house one or more application systems. The number depends on the complexity of the application systems and installation processing policy. The CICS TS is recommended to run as an MVS-started tasks in production environment; however, it can also be run as an OS/390 job during the testing period. The JCL procedure shown in Figure 14 on page 61 is one we set up for this project and is listed for reference only. It should be changed to meet installation naming convention.

```
*place JCL in member     SYS1.PROCLIB(CICSD)

//CICSD    PROC START='INITIAL',
//            INDEX='CICS.SJCICSD',
//            INDEX1='CICS.SJCICSD',
//            SYSIDNT=PAAD,    *** PAA1, PAA2, PAA3, PAA4
//            REG='0M',
//            OUTC='*'
//CICS520  EXEC PGM=DFHSIP,REGION=&REG,TIME=1440,
//            PARM=('START=&START',
//            'APPLID=SCSC&SYSIDNT',
//            'GRPLIST=(DFHLIST)',
//            'SYSIDNT=CICD',
//            'SYSIN')
//STEPLIB  DD  DISP=SHR,DSN=&INDEX1..SDFHAUTH
//         DD  DSN=IMS.SJIMSD.RESLIB,DISP=SHR
//SYSABEND DD  SYSOUT=&OUTC
//SYSIN    DD  DSN=&INDEX..PARMLIB(OVERRIDE),DISP=SHR
//DFHRPL   DD  DSN=&INDEX1..SDFHLOAD,DISP=SHR
//         DD  DSN=&INDEX..TABLOAD,DISP=SHR
//         DD  DSN=CEE.SCEECICS,DISP=SHR
//         DD  DSN=CSS.SCEERUN,DISP=SHR
//DFHTEMP  DD  DSN=&INDEX..DFHTEMP,DISP=SHR
//DFHINTRA DD  DSN=&INDEX..DFHINTRA,DISP=SHR
//DFHAUXT  DD  DSN=&INDEX..DFHAUXT,DISP=SHR,DCB=BUFNO=5
//DFHBUXT  DD  DSN=&INDEX..DFHBUXT,DISP=SHR,DCB=BUFNO=5
//DFHCMACD DD  DSN=&INDEX1..DFHCMACD,DISP=SHR,DCB=BUFNO=5
//DFHCXRF  DD  SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//LOGUSR   DD  SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR   DD  SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//PLIMSG   DD  SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//COUT     DD  SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG   DD  SYSOUT=&OUTC
//CEEOUT   DD  SYSOUT=&OUTC
//DFHDMPA  DD  DSN=&INDEX..DFHDMPA,DISP=SHR
//DFHDMPB  DD  DSN=&INDEX..DFHDMPB,DISP=SHR
//DFHGCD   DD  DSN=&INDEX..DFHGCD,DISP=OLD
//DFHLCD   DD  DSN=&INDEX..DFHLCD,DISP=SHR
//DFHCSD   DD  DSN=&INDEX..DFHCSD,DISP=SHR
```

Figure 14. CICS TS Online Region Sample JCL

The listing is required for the system environment only. Add your application program's load-library as required.

7.4 Constructing CICS TS Initial Modules

The following are the procedures to be able to bring up both IMS DBCTL and CICS TS online regions for application. Some procedures are not necessary but are highly recommended to make operations much easier.

7.4.1 Install SIT Module for CICS TS

There are two ways to construct the CICS TS SIT module. The first is to use a sample SIT module that came with the product and overwrite it with parameters appropriate for the application region. The second method is to code the module macro, then compile and link to store in an APF library. This may be useful in a production environment to avoid an operation accident. For testing, a simple default SIT module is good enough; the one we used had already been placed in the CICS APF library, SDFHAUTH. The statements to code this are given in Appendix B, "Resources Needed by Initial Installer" on page 239.

Care should be taken even when the user wants to code his own SIT module. There are several parameters that are not allowed to be coded when the DFHSIT macro is used; they must be entered as overwrite parameters. Figure 15 shows such parameters.

```
        CDSASZE       OFFSITE
        CHKSTRM       PRVMOD
        CHKSTSK       RDSASZE
        ECDSASZE      SDSASZE
        ERDSASZE      SIT
        ESDSASZE      START
        EUDSASZE      UDSASZE
        NEWSIT
```

Figure 15. CICS TS SIT Parameters Cannot Be Coded with the DFHSIT Macro

7.4.2 Install Start-up PLT Module for CICS TS

The PLT module is part of the CICS initial processing. One of its function is to connect to other subsystems automatically. Figure 16 shows the PLT macro coding; the load module is placed in the CICS SDFHLOAD library. The module is set to connect to IMS DBCTL subsystem at the CICS initial time. The JCL to assemble this is given in Appendix B, "Resources Needed by Initial Installer" on page 239.

```
*  place statements in member CICS.SJCICSD.TABSRC(PLTPI)

        DFHPLT TYPE=INITIAL,SUFFIX=PI
        DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
        DFHPLT TYPE=ENTRY,PROGRAM=DFHDBCON
        DFHPLT TYPE=FINAL
        END
```

Figure 16. Start-up PLT Statements

There are also new SIT parameters on CICS TS to start the DBCTL (and other subsystems as well) with automatic connection during initialization. The parameters are:

DB2CONN Start the DB2 connection automatically.

DBCTLCON Start the DBCTL connection automatically.

MQCONN Start the CICS MQSeries connection automatically.

WEB Start the CICS Web interface automatically.

When these parameters are coded as part of the SIT overwrite statements, then PLT is no longer required.

7.5 CICS TS Journaling and Logging

One of the major differences between CICS VSE and CICS TS is that the CICS log manager replaces the journal control function. With OS/390 V2R4, CICS TS V1R2 offers three types of journaling for the user community:

- **Coupling facility logging**. This is the most sophisticated CICS logging that works in the OS/390 sysplex environment. It requires a coupling facility and the CICS log stream can contain data from multiple systems, allowing a system logger application to merge data from systems across the Parallel Sysplex.
- **DASD-only logging**. A DASD-only log stream connects to one MVS image at a time. Multiple applications from the same system can access the log stream simultaneously. As a log stream fills log data sets on DASD, the system logger automatically allocates new log data sets for the log stream.
- No logging.

Since most VSE users who are migrating to the OS/390 will not set up a Parallel Sysplex environment immediately, we selected DASD-only logging for our sample migration.

7.5.1 Defining the MVS System Logger

The CICS system log is implemented as two MVS system logger log streams. One stream is the primary system log stream, DFHLOG, which holds data for most normal (short-lived) in-flight units of work. The other stream is the second system log stream, DFHSHUNT, which holds information for UOWs that are not short-lived, typically UOWs that cannot complete because of back-out failures, or because they are designed as long-running tasks that issue infrequent sync points.

The two log streams can be defined as DASD-only log streams (or as coupling facility log streams). Figure 17 on page 64 shows our definition of DFHLOG and DFHSHUNT with DASD-only log streams.

```
//CHENSA  JOB  (999,POK),CHENS,CLASS=A,NOTIFY=&SYSUID,
//        MSGLEVEL=(1,0),MSGCLASS=X
//*
//LOGDEFN  EXEC PGM=IXCMIAPU
//STEPLIB  DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//***********************************************************
//*                                                         *
//* DEFINE DASD-ONLY LOG STREAMS FOR CICS SYSTEM LOG.       *
//*                                                         *
//***********************************************************
//SYSIN    DD *
  DATA TYPE(LOGR) REPORT(YES)
  DEFINE LOGSTREAM NAME(CICS.SJCICSD.DFHLOG)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000) STG_SIZE(3000)
                   LOWOFFLOAD(60) HIGHOFFLOAD(95)

  DEFINE LOGSTREAM NAME(CICS.SJCICSD.DFHSHUNT)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000) STG_SIZE(500)
                   LOWOFFLOAD(0) HIGHOFFLOAD(80)
/*
```

Figure 17. Sample CICS DASD-only Logging Definition

7.5.1.1 Log Data Sets

The size for DASD log data sets for each log stream is specified on the LS_SIZE parameter in the log stream definition. When the LS_SIZE parameter is omitted, MVS LOGR uses the size defined in the SMS data class for the log stream data sets. Alternatively, if SMS is not available, MVS LOGR uses dynamic allocation rules for allocating data sets.

The log data sets should be as large as the installation can afford. This minimizes the number of log data sets required to represent a log stream.

Figure 18 on page 65 shows the output of our logger job listed in Figure 17.

```
LOGSTREAM NAME(CICS.SJCICSD.DFHLOG) STRUCTNAME() LS_DATACLAS()
          LS_MGMTCLAS() LS_STORCLAS() HLQ(IXGLOGR) MODEL(NO) LS_SIZE(0)
          STG_MGMTCLAS() STG_STORCLAS() STG_DATACLAS() STG_SIZE(3000)
          LOWOFFLOAD(60) HIGHOFFLOAD(95) STG_DUPLEX(YES) DUPLEXMODE(UNCOND)
          RMNAME() DESCRIPTION() RETPD(0) AUTODELETE(NO)
          DASDONLY(YES)
          MAXBUFSIZE(64000)

LOGSTREAM NAME(CICS.SJCICSD.DFHSHUNT) STRUCTNAME() LS_DATACLAS()
          LS_MGMTCLAS() LS_STORCLAS() HLQ(IXGLOGR) MODEL(NO) LS_SIZE(0)
          STG_MGMTCLAS() STG_STORCLAS() STG_DATACLAS() STG_SIZE(500)
          LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(YES) DUPLEXMODE(UNCOND)
          RMNAME() DESCRIPTION() RETPD(0) AUTODELETE(NO)
          DASDONLY(YES)
          MAXBUFSIZE(64000)
```

Figure 18. DASD-only Log Definition Report

When the space occupied by a log stream reaches the installation-defined threshold. The MVS logger off-loads the log blocks from the log-stream data space areas to DASD make room for new log records. Refer to *CICS Transaction Server for OS/390 Installation Guide, GC33-1681* for detailed information.

7.5.1.2 Staging Data Sets

For DASD-only log streams, staging data sets are a required part of the system logger configuration. The MVS system logger automatically duplexes data to the staging data set for the system at the same time it writes the data to local storage buffers.

The STG_SIZE parameter in the log stream definition is the size of the staging data sets. Otherwise, MVS uses the maximum VSAM linear data set size.

7.5.2 Defining the CICS Log Manager

The CICS log manager replaces the journal control function of earlier CICS releases. To map the MVS log stream to CICS, the user needs to define a CICS resource called *JOURNALMODEL* in the CSD.

The CICS system log requires two JOURNALMODEL definitions, one with a JOURNALNAME parameter of DFHLOG and one with a JOURNALNAME parameter of DFHSHUNT. For both JOURNALMODEL definitions, the TYPE parameter must be MVS.

The STREAMNAME parameter associates the CICS definition with the log streams defined in the DASD-only MVS logger. The sample batch definitions are shown in Figure 19 on page 66. Users can also employ RDO to define the JOURNALMODEL.

```
//DFHCSDUP JOB (999,POK),CHENS,CLASS=A,MSGCLASS=T,
//         NOTIFY=&SYSUID
//*
//STEP1    EXEC PGM=DFHCSDUP,REGION=2M
//STEPLIB   DD  DSN=CICSTS12.CICS.SDFHLOAD,DISP=SHR
//DFHCSD    DD  DSN=CICS.SJCICSD.DFHCSD,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD *
 DEFINE JOURNALMODEL(SYSLOG)
     GROUP(LOG62)
     DESCRIPTION('DFHLOG LOGSTREAM EXPLICIT DEFINITION')
     JOURNALNAME(DFHLOG)
     TYPE(MVS)
     STREAMNAME(CICS.SJCICSD.DFHLOG)
 DEFINE JOURNALMODEL(SYSSHUNT)
     GROUP(LOG62)
     DESCRIPTION('DFHSHUNT LOGSTREAM EXPLICIT DEFINITION')
     JOURNALNAME(DFHSHUNT)
     TYPE(MVS)
     STREAMNAME(CICS.SJCICSD.DFHSHUNT)
//
```

Figure 19. Sample CICS System Log Definitions

7.5.3 Defining User Journals and Autojournals

The CICS TS allows numeric journal identifiers for:

- File control autojournaling, as specified in FILE resource definitions (or on DFHFCT macro entries).
- Terminal control autojournaling, as specified in PROFILE resource definitions.
- Forward recovery logging, as specified in FILE resource definitions.
- User journaling using API journal commands, such as the EXEC CICS WRITE JOURNALNAME command.

The setup of those DFHJxx journals is no different from the setup of a CICS system log. Users must define the MVS log stream for the journal as follows (see Figure 17 on page 64):

```
DEFINE LOGSTREAM NAME(CICS.SJCICSD.DFHJnn)
                 DASDONLY(YES)
                 MAXBUFSIZE(64000) STG_SIZE(3000)
                 LOWOFFLOAD(60) HIGHOFFLOAD(95)
```

Then, define a JOURNALMODEL as the CICS system defininition (CSD), as follows:

```
DEFINE JOURNALMODEL(DFHJnn)
    GROUP(LOG62)
    DESCRIPTION('AUTOJOURNAL DEFINITION')
    JOURNALNAME(DFHJnn)
    TYPE(MVS)
    STREAMNAME(CICS.SJCICSD.DFHJnn)
```

Unlike the journal control table, users need not define a journal model for every journal that CICS uses. Instead, the above model can be defined in a generic model definition that describes the mapping to log stream names for the majority of the CICS journals.

For other log management functions, refer to the ITSO redbook and other publications listed at the beginning of this chapter.

7.5.4 Testing Environment

For the initial testing environment, the CICS TS region can be started without logging, which implies no possibility of recovery. Figure 20 shows the example.

```
*  place JCL in member CICS.SJCICSD.DBDC.JCL(NOLOGS)

//INIT     EXEC PGM=DFHCSDUP,REGION=4M
//STEPLIB  DD DSN=CICS.SJCICSD.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICS.SJCICSD.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE JOURNALMODEL(DFHJ01) JOURNALNAME(DFHLOG) TYPE(DUMMY)
         GROUP(LOGDD)
  DEFINE JOURNALMODEL(DFHLOG) JOURNALNAME(DFHLOG) TYPE(DUMMY)
         GROUP(LOGDD)
  ADD GROUP(LOGDD) LIST(TEMPDLST)
/*
//
```

Figure 20. Sample JCL to Define Dummy Log

7.6 IMS Application Region Generation

There are three types of IMS generation the user needs to know about: full generation; nucleus generation, and MODBLKS generation.

7.6.1 IMS Full Generation

This is a several-stage process, which the user need to run only during initial installation and when applying system maintenance that affects the IMS online region nucleus. Full generation includes all the latest maintenance functions and builds in the latest application and database definitions. You need a cold start of IMS to install a full generation. The IMS full generation is described in B.8, “The IMS DBCTL Full Generation Processing” on page 250.

Application system customization processing need not include this type of generation.

7.6.2 IMS Nucleus Generation

Like full generation, this process needs to run only once during the customization work. The nucleus is relinked automatically when any system maintenance is applied in a SMP control environment. The procedure is the same as for full generation, but fewer jobs are generated by the first stage of two-stage generation processing.

7.6.3 IMS MODBLKS Generation

After the IMS DBCTL application region is customized, and the user wants to amend the application or database definitions, he or she should then run a MODBLKS generation. The generation does not include any new maintenance functions but application resources, and can be installed while IMS is running. The MODBLKS generation is described in full in Chapter 8, “Application Migration” on page 81.

7.7 Customizing IMS/ESA DBCTL Application Region JCL

IMS DBCTL is a multiple MVS address space region and their JCLs are, in general, created by the nucleus-generation processing. The user may have to adjust the JCL, such as adding a new application library, and so on, to meet user needs.

We recommend that an IMS DBCTL region house as many application-system databases as possible to avoid MVS common storage area constraints. IMS DBCTL is an MVS multitask on-line region that contains a minimum of three address spaces. Other JCL procedures are also associated with the IMS/ESA DBCTL region:

- IMS control region.
- IMS DBRC region.
- IMS DL/I databases region.
- IMS BMP regions procedure.
- IMS off-line batch region procedure.

We recommend running the IMS DBCTL region as an MVS-started tasks in the production environment. However, it can also be run as an OS/390 job during the testing period. The DBRC and DL/I database regions are always MVS-started tasks that are started by the DBCTL region as part of initialization processing. The other region procedures could be run either as batch jobs or started tasks. The log archive job can be run manually or as an MVS-started task during automatic log switch.

7.7.1 IMS/ESA DBCTL Online Region JCL

The IMS DBCTL region JCL is shown in Figure 21 on page 69. Its primary function is communication with the CICS region.

```
*place JCL in member     SYS1.PROCLIB(SJIMSD)

//SJIMSD  PROC  RGN=0M,
//             SOUT='*',
//             DPTY='(14,15)',
//             SYS1='SJIMSD.',
//             RGSUF=SJD,
//             SUF=D,
//             AUTO=N,
//             PARM2=
//*******************************************************************
//* ESA.SYS1.PROCLIB(SJIMSD)
//*-----------------------------------------------------------------*
//*******************************************************************
//*
//IMS     EXEC PGM=DFSMVRC0,REGION=&RGN,DPRTY=&DPTY,
// PARM=('DBC,&RGSUF,SUF=&SUF.,AUTO=&AUTO,&PARM2')
//*
//STEPLIB  DD DSN=IMS.&SYS1.USERLIB,DISP=SHR
//         DD DSN=IMS.&SYS1.RESLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.&SYS1.PROCLIB,DISP=SHR
//*
//*********** DASD  LOGGING DD CARDS *************
//*
//* LOGGING  dataset ALLOCATIONS DYNAMIC & PICKED UP FROM USERLIB
//*
//***** MESSAGE QUEUE DD CARDS *****  OMIT FOR A DBCTL REGION
//*
//*
//IMSACBA  DD DSN=IMS.&SYS1.ACBLIBA,DISP=SHR
//IMSACBB  DD DSN=IMS.&SYS1.ACBLIBB,DISP=SHR
//MODBLKSA DD DSN=IMS.&SYS1.MODBLKSA,DISP=SHR
//MODBLKSB DD DSN=IMS.&SYS1.MODBLKSB,DISP=SHR
//MODSTAT  DD DSN=IMS.&SYS1.MODSTAT,DISP=SHR
//*
//******** SYSTEM REQUIRED DD CARDS **************
//*
//SYSMDUMP DD DSN=IMS.&SYS1.SYSMDUMP,DISP=SHR
//IMSRDS   DD DSN=IMS.&SYS1.RDS,DISP=SHR
//MATRIXA  DD DSN=IMS.&SYS1.MATRIXA,DISP=SHR
//MATRIXB  DD DSN=IMS.&SYS1.MATRIXB,DISP=SHR
//PRINTDD  DD SYSOUT=&SOUT
//IMSIRD   DD SYSOUT=(A,INTRDR)
```

Figure 21. Sample JCL for IMS/ESA DBCTL Control Region

The SYSMDUMP statement on the listing produces a dump data set used by the MVS IPCS subsystem for online dump analysis. Users can replace this statement by either a SYSUDUMP or SYSABEND statement to produce a format dump stored either on DASD or on paper.

7.7.2 IMS/ESA Database Address Space JCL

This is the second address space of the IMS/ESA DBCTL region. The JCL is shown in Figure 22. This address space is responsible for all databases access.

```
* place JCL in member     SYS1.PROCLIB(DLISAS)

//DLISAS  PROC RUN=SJIMSD
//IEFPROC EXEC PGM=DFSMVRC0,
//        REGION=0M,
//        PARM=(DLS,IMSD),
//        PERFORM=7,TIME=1440,DYNAMNBR=300
//*********************************************
//*
//STEPLIB  DD  DSN=IMS.&RUN..USERLIB,DISP=SHR
//         DD  DSN=IMS.&RUN..RESLIB,DISP=SHR
//PROCLIB  DD  DSN=IMS.&RUN..PROCLIB,DISP=SHR
//*****************ACBLIB**********************
//*
//* THE SPECIFICATION OF THE ACBLIB datasetS
//* IN THE DLI/SAS REGION PROCEDURE MUST
//* CORRESPOND EXACTLY WITH THE SPECIFICATION
//* IN THE CONTROL REGION JCL
//*
//IMSACBA  DD  DISP=SHR,DSN=IMS.&RUN..ACBLIBA
//IMSACBB  DD  DISP=SHR,DSN=IMS.&RUN..ACBLIBB
//SYSMDUMP DD  DISP=SHR,DSN=IMS.&RUN..DLISAS.SYSMDUMP
//*
//*   ALL DATABASES DEFINED VIA DYNAMIC ALLOCATION
//*
//************ DATA BASE DD CARDS ***************
//*
//* USER MAY OPTIONALLY SUPPLY THE DD STATEMENTS
//* FOR THE ON-LINE DATA BASES TO BE
//* INSERTED HERE PRIOR TO ATTEMPTING
//* AN ON-LINE SYSTEM EXECUTION USING
//* THIS PROCEDURE.
//* IF NO DD STATEMENTS ARE SUPPLIED FOR
//* A DATA BASE, IMS/VS ASSUMES THAT THIS
//* DATA BASE HAS BEEN DESCRIBED THROUGH
//* THE DFSMDA MACRO.
//*
```

Figure 22. Sample JCL for IMS DL/I Region

The JCL procedure in Figure 22 implies that all databases will be allocated dynamically by means of load modules using the DFSMDA macro. Add DD statements as needed if you do not use dynamic allocation.

7.7.3 IMS DBCTL DBRC Region JCL

This is the third address space of the IMS DBCTL. Its procedure is shown in Figure 23, and the primary function is database recovery. The PROCLIB statement evokes a library that stores model JCL procedures (see 7.7.4, "IMS DBCTL Log Archive Procedures"). Once again, the database control files RECON data sets are accessed through dynamic allocation.

```
* place JCL in member     SYS1.PROCLIB(DBRC)

//DBRC    PROC RUN=SJIMSD
//IEFPROC EXEC PGM=DFSMVRC0,REGION=1000K,
// PERFORM=7,TIME=1440,PARM=(DRC,IMSD)
//**********************************************
//*
//STEPLIB  DD  DSN=IMS.&RUN..USERLIB,DISP=SHR
//         DD  DSN=IMS.&RUN..RESLIB,DISP=SHR
//PROCLIB  DD  DSN=IMS.&RUN..PROCLIB,DISP=SHR
//JCLOUT   DD  SYSOUT=(A,INTRDR)
//JCLPDS   DD  DSN=IMS.&RUN..JCLLIB,DISP=SHR
//SYSMDUMP DD  DISP=SHR,DSN=IMS.&RUN..DBRC.SYSMDUMP
```

Figure 23. Sample JCL for IMS DBRC Region

7.7.4 IMS DBCTL Log Archive Procedures

Figure 24 on page 72 is not a real JCL procedure but a JCL model procedure. It uses the online region to archive the IMS log when the IMS log is full or meets the archive criteria. IMS uses this model to assemble a job, then submit it to OS/390 internally. Alternatively, if the archive is set to manual operation, IMS displays a message alerting the operator to the need for archiving. The operator then submits a DBRC log archive job to OS/390 or issues a DBRC command, all based on this model, to trigger archive processing.

```
*  place JCL in member IMS.SJIMSD.JCLLIB(ARCHJCL)

//ARC%SSID JOB (999,ITSO),'ARCHIVE %SSID IMSD',
//         CLASS=B,MSGCLASS=T,MSGLEVEL=(1,1),
//         TIME=(1440),REGION=6M
//*
//*                 JCL FOR ARCHIVE UTILITY
//*                 -----------------------
//*
//*  JOB GENERATED BY DBRC (STARTED TASK OR BATCH UTILITY) FROM
//*  SKELETON JCL IN IMSVS.PROCLIB(ARCHJCL).
//*
//*  RECONS DYNAMICALLY ALLOCATED FROM MDS MEMBERS IN IMS.*.USERLIB
//*
//*  AMENDMENT RECORD
//*
//*-------------------------------------------------------------------
//* STEP: ARCHIVE - RUN IMS UTILITY TO ARCHIVE OLDS TO SLDS
//*-------------------------------------------------------------------
//*
//* SET MIDDLE LEVEL QUALIFIER FOR dataset NAMES.
//*
//      SET SYS1=SJIMSD
//ARCHIVE   EXEC PGM=DFSUARCO,PARM='%SSID,DBRC=YES'
//*
//STEPLIB   DD DISP=SHR,DSN=IMS.&SYS1..USERLIB   DYNAL FOR RECONs
//          DD DISP=SHR,DSN=IMS.&SYS1..RESLIB    NEED A 6.1 RESLIB
//SYSPRINT  DD DISP=(NEW,PASS),DSN=&&LIST,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6118)
//SYSUDUMP  DD SYSOUT=*
//*
%SELECT  OLDS(%SSID,(%DDNAMES))
//%OLDSDDN  DD DSN=%OLDSDSN,DISP=SHR,DCB=BUFNO=40
%ENDSEL
//*
//DFSSLOGP  DD DSN=IMS.&SYS1..SLDSP.D%ARDATE.T%ARTIME.V%ARVERS,
//             SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA,DISP=(,CATLG),
//          DCB=(IMS.&SYS1..OLPOO,LRECL=22520,BLKSIZE=30720,RECFM=VBS)
//SYSIN     DD *
SLDS FEOV(08000)
//*--------------------------------------------------------------------
//* PRINT ARCHIVE LISTING IF THE PREVIOUS STEP DIDN'T
//*--------------------------------------------------------------------
//PRINT    EXEC PGM=IEBGENER
//SYSUT1    DD DISP=(SHR,PASS),DSN=&&LIST
//SYSUT2    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//*--------------------------------------------------------------------
//* ALWAYS DELETE THE LISTING dataset
//*--------------------------------------------------------------------
//DELETE   EXEC PGM=IEFBR14
//LISTIN    DD DISP=(OLD,DELETE),DSN=&&LIST
```

Figure 24. JCL for IMS to Archive OLDS

7.8 Constructing DBCTL Process Modules

A few of the internal process modules required customization.

7.8.1 Construct DRA Module

The DRA module is the CICS TS and IMS/ESA interface module. The macros are shipped with IMS/ESA. Use IMS assembly JCL to compile and link this module and place it in the CICS TS APF library for CICS TS application region use. (A sample job is shown in Figure 131 on page 249)

Once the module is created, the interface can be activated by the INITPARM parameter in SIT override list as:

```
INITPARM=(DFHDBCON='0D,IMSD')
```

where

0D

is the suffix of the DRA module, DFSPZPxx, and

IMSD

is the DBCTL region ID. A CICS TS online region can access as many DBCTL as needed but one at a time. On the other hand, one DBCTL online can be connected to many CICS TS regions concurrently.

7.8.2 Create Dynamic Allocation Definitions

Many IMS data sets can be allocated dynamically. The advantages are a lot less JCL to code, and that you always have a consistent set of data sets. It also allows you to remove the data sets from IMS while it is running. Users would commonly use dynamic allocation for the following IMS data sets:

OLDS IMS log data sets include:

- OLPnn. Primary log data sets.
- OLSnn. Secondary log data sets.

WADS IMS log write-ahead data sets.

RECONs. IMS DBRC database control data sets.

DFSTRA0T IMS trace data set on tape.

IMSLOGR Input log data set for extended restart.

IMSMON Use by IMS monitor.

Databases Application database data sets

A list of the JCL and macro coding needed for this can be seen in Appendix B, “Resources Needed by Initial Installer” on page 239.

7.8.3 Constructing the Initial ACB Library

The ACB library existed only in the IMS environment, not CICS local DL/I. The following steps must be run to construct the ACB library for the first time:

1. Run the DBDGEN.
2. Run the PSBGENs.
3. Run the ACBGEN.
4. Run the ACB copy.

Run the jobs as described below:

1. Run the DBDGEN with all application DBDs. The source statements for this DBDGEN are coded by the application developer. The JCL to assemble DBD is shown in Figure 25.

```
* place statements in member IMS.SJIMSD.DBDC.JCL(DBDGDBMP)

//DBDGEN   PROC MBR=TEMPNAME,SOUT='*',RGN=1M,SYS2='SJIMSD.'
//*
//C        EXEC PGM=ASMA90,REGION=&RGN,PARM='OBJECT,NODECK'
//SYSLIB    DD DSN=IMS.&SYS2.MACLIB,DISP=SHR
//SYSLIN    DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1),RLSE),
//             DCB=(BLKSIZE=0,RECFM=FB,LRECL=80)
//SYSPRINT  DD SYSOUT=&SOUT,DCB=BLKSIZE=6144,SPACE=(TRK,(1,1),RLSE)
//SYSUT1    DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(25,10))
//SYSIN     DD DISP=SHR,DSN=IMS.&SYS2.DBDSLIB(&MBR)
//*
//L        EXEC PGM=IEWL,PARM='XREF,LIST',COND=(0,LT,C),REGION=2M
//SYSLIN    DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//SYSPRINT  DD SYSOUT=&SOUT,DCB=BLKSIZE=6144,SPACE=(TRK,(1,1),RLSE)
//SYSLMOD   DD DISP=SHR,DSN=IMS.&SYS2.DBDLIB(&MBR)
//SYSUT1    DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//             SPACE=(1024,(100,10),RLSE),DISP=(,DELETE)
//      PEND
//*
//S1    EXEC DBDGEN,MBR=DFHDBMP
/*
```

Figure 25. Sample DBD Generation Procedure

2. Run the PSBGEN with all application PSBs. The source statements should also be coded by application developers and the JCL to assemble PSB is similar, with Figure 25 just replacing DBD with PSB.
3. Run the ACBGEN. Both all application DBDGEN and PSBGEN must run successfully without error before this job can be run. Figure 26 on page 75 shows the JCL procedure.

```
place statements in member IMS.SJIMSD.DBDC.JCL(ACBINIT)

//BOLL   PROC SOUT='*',COMP=,SYS1='SJIMSD.'
//*
//G      EXEC PGM=DFSRRC00,PARM='UPB,&COMP'
//STEPLIB  DD DISP=SHR,DSN=IMS.&SYS1.RESLIB
//SYSPRINT DD SYSOUT=&SOUT
//IMS      DD DSN=IMS.&SYS1.PSBLIB,DISP=SHR
//         DD DSN=IMS.&SYS1.DBDLIB,DISP=SHR
//IMSACB   DD DSN=IMS.&SYS1.ACBLIB,DISP=OLD
//SYSUT3   DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4   DD UNIT=SYSDA,SPACE=(256,(100,100)),
//            DCB=KEYLEN=8
//COMPCTL  DD DISP=SHR,
//         DSN=IMS.&SYS1.PROCLIB(DFSACBCP)
//    PEND
//*
//    EXEC BOLL,COMP=POSTCOMP
//SYSIN DD *
    BUILD PSB=ALL

 or BUILD PSB=DFHDBMP if you only want to build this one only.
```

Figure 26. Sample ACBGEN Procedure

4. Run the offline ACB copy. ACBs are normally copied using online changes; the offline job is documented in Appendix B, “Resources Needed by Initial Installer” on page 239.

7.9 Constructing Initial Parameter Lists

There are a number of parameter lists which the user should find on set up for both CICS and IMS application regions. Some of these need to be set up only once; these are described in Appendix B, "Resources Needed by Initial Installer" on page 239. Others need to be amended either to tune the system, or to add extra functions. These are described in this chapter.

7.9.1 Initialize the CICS TS CSD

Figure 27 shows the sample JCL for an MVS job that formats the CICS CSD, and installs all the IBM-supplied definitions.

```
*  place JCL in member CICS.SJCICSD.DBDC.JCL(CSDINIT)

//INIT     EXEC PGM=DFHCSDUP,REGION=4M
//STEPLIB  DD DSN=CICS.SJCICSD.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICS.SJCICSD.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
           INITIALIZE
           LIST ALL OBJECTS
           VERIFY
/*
//
```

Figure 27. Sample JCL to Initialize the CSD

7.9.2 Create CICS Start-up Overrides Member

The user must create a list of start-up parameters that override the default SIT module. A suitable list is shown in Figure 28.

```
* place in member CICS.SJCICSD.DBDC.JCL(OVERRIDE)

    PARMERR=IGNORE
    APPLID=SCSCPAAD         VTAM APPLID
    SIT=D0
    GRPLIST=(DFHLIST,TEMPDLST)
    SYSIDNT=CICD
    MNSUBSYS=CICD
    LPA=NO
    DBCTLCON=YES
    DCT=NO
    FCT=NO           MAY NEED TO DEFINE.  - CREATE DUMMY.
    MCT=NO
    PDIR=NO          MAY NEED TO DEFINE FOR REMOTE PSBS
    PLTPI=PI         NEED PLT FOR INIT PROGRAMS.
    PLTPISEC=NONE    NO SECURITY CHECKS DURING PLT PROCESSING
    PLTSD=NO         WILL NEED PLT FOR CLOSEDOWN PROGRAMS
    SRT=NO           SRT WILL NEED TO BE CODED.
    TCT=NO
    TST=NO
    SECPRFX=YES
    SEC=NO           NO SECURITY DURING TESTING
    XLT=NO
    XCMD=NO
    XDCT=NO
    XFCT=NO
    XJCT=NO
    XPCT=NO
    XPPT=NO
    XPSB=NO
    XTRAN=NO
    XTST=NO
    XUSER=NO
    AUXTRSW=NEXT     AUX TRACE dataset SWITCHES ONCE
    CICSSVC=216      CICS TYPE 3 SVC
    SRBSVC=215       CICS TYPE 6 SVC
    HPO=YES          VTAM  AUTHORIZED PATH ROUTING
    CSDBKUP=DYNAMIC
    CSDDISP=SHR
    CSDLSRNO=1
    CSDRLS=NO        DON'T USE RLS FOR CSD
    CSDSTRNO=8
    DLI=NO           NO LOCAL DL/I
```

Figure 28 (Part 1 of 2). Sample CICS Start-Up Parameters

```
DFLTUSER=CICDSTC            IF YOU DON'T SIGN ON GET default ID
DSHIPIDL=0000500
DSHIPINT=0001000
DUMPSW=NEXT
GMTEXT='CICS/ESA FOR DBCTL AT ITSO, SAN JOSE'
INITPARM=(DFHDBCON='0D,IMSD')
ISC=YES
LGNMSG=YES
MN=OFF
MROBTCH=3
MROLRM=YES
MXT=50
NEWSIT=YES
OPERTIM=1800
PGAICTLG=NONE
PGAIEXIT=DFHPGADX
PGAIPGM=ACTIVE
PGCHAIN=CHN&
PGCOPY=COP&
PGPURGE=PUR&
PGRET=RET&
PRINT=PA1
PSBCHK=NO
SNSCOPE=NONE
SUBTSKS=1
TD=(12,10)
TDINTRA=NOEMPTY
TS=(,12,10)
VTPREFIX=
XDB2=NO
.END
```

Figure 28 (Part 2 of 2). Sample CICS Start-Up Parameters

7.9.3 Create IMS Start-up Overrides Member

A start-up parameters list of overrides to the defaults supplied in the IMS GEN is also needed. A suitable list is shown in Figure 29.

```
* place in member IMS.SJIMSD.PROCLIB(DFSPBSJD)

    CMDMCS=R,
    SSM=NUL,
    RES=N,
    CRC=>,
    ARMRST=N,
    PST=10,
    SRCH=0,
    PSB=,
    DMB=6K,
    CIOP=,
    WKAP=060,
    PSBW=36,
    DBWP=08,
    SUF=D,
    FIX=DC,
    PRLD=,
    VSPEC=IB,
    SOD=,
    BSIZ=,
    OTHR=,
    DBFX=,
    DBBF=,
    MSDB=,
    FMTO=T,
    AUTO=N,
    IMSID=IMSD,
    ISIS=0,
    LGNR=,
    IRLM=N,
    IRLMNM=,
    WADS=D,
    ARC=01,
    UHASH=,
    QTU=,
    DBRCNM=DBRC,
    DLINM=DLISAS,
    CSAPSB=10,
    DLIPSB=20,
    PRDR=IMSDRDR,
    EPCB=,
    FPWP=,
    SPM=,
    PIMAX=20,
    PIINCR=1,
    AOIS=S
```

Figure 29. Sample IMS Start-Up Parameters

Several override parameter lists are also needed by the IMS DBCTL. These are usually created by the IMS full or nucleus generation and modified as needed when application resources are added to the online region.

7.9.4 Create IMS VSAM Parameters

Figure 30 shows a common use of IMS VSAM parameters needed by the DBCTL online region. Refer to IMS product manual *IMS/ESA, Installation Volume I, GC26-8736* for a detailed description of these parameters.

```
* place in member IMS.SJIMSD.PROCLIB(DFSVSMSB)

    OPTIONS,VSAMFIX=IOB
    OPTIONS,VSAMPLS=LOCL
    VSRBF=0512,20
    VSRBF=1024,20
    VSRBF=2048,20
    VSRBF=4096,20
    VSRBF=8192,20
    VSRBF=12288,20
    VSRBF=16384,20
    IOBF=(1024,8,N,Y)
    IOBF=(2048,4,N,Y)
    IOBF=(4096,4,N,Y)
    IOBF=(8192,4,N,Y)
    IOBF=(12288,5,N,Y)
    OLDSDEF OLDS=(00,01,02,03,04),BUFNO=005,MODE=DUAL
    WADSDEF WADS=(0,1,2)
    CFNAMES,CFIRLM=IRLMT1,CFVSAM=IRLMVSAMT1,CFOSAM=IRLMOSAMT1
```

Figure 30. Sample IMS VSAM Parameters

Chapter 8. Application Migration

This chapter shows the application side of the migration. Once the system environment that houses the application system has been built (described in Chapter 7, "System Environment Customization" on page 57), the application migration just adds additional application-related items to the processing tables, JCL procedures, control blocks, and adjusts online-region processing parameters.

8.1 Summary of Application Migration

Following the procedures in 6.9, "Project Migration" on page 45, we begin to migrate applications from VSE to OS/390 environment. In this section, we give an overview of major tasks. The remaining sections of this chapter document the actual work we performed here at ITSO San Jose center.

8.1.1 Pre Migration Tasks

Before migration begins, these two tasks must be completed:

- Establish CICS TS and IMS/ESA DBCTL in the OS/390 environment.
- Migrate CICS tables PCT and PPT to online definitions on VSE.

8.1.2 Main Migration Tasks

The following are not necessarily in order, but some steps do need to be completed before others, as noted in the related sections:

1. Copy the CICS CSD from CICS/VSE to CICS/TS.
2. Copy all current CICS tables from VSE to OS/390 as source.
3. Copy all current DL/I DBDs and PSBs from VSE to OS390 as source.
4. Migrate CICS tables TCT and FCT to online definitions on OS390.

   The user must migrate FCT VSAM definitions and TCT VTAM and console definitions to RDO. However, some definitions cannot be migrated, such as BDAM files, or TCAM or sequential terminals. The user must retain such resources in FCT or TCT if the user does have any of these. Also PDIR and DDIR will be processed, largely for conversion to IMSGEN input.
5. Recode the SRT to change VSE abend codes to MVS codes, and assemble and link them to the TABLOAD library.
6. Assemble and link other CICS tables if they appear. For example, CLT, DCT, MCT, PLTs, TLT, TST and XLT must be assembled into a TABLOAD library.
7. Respond as appropriate to the fact that JCT, ALT and NLT are no longer used.
8. Migrate logging from CICS journals to the OS/390 MVS system logger. DL/I logging is now done by IMS DBCTL. If the user is using the CICS logs only for recovery of the DL/I database in his or her VSE environment, then it does not need any CICS logging in the new IMS DBCTL environment, since IMS knows independently of CICS whether a transaction succeeded or not.
9. Respond to the fact that DLZACT no longer used under IMS DBCTL.
10. Recode DBDs and assemble into IMS DBDLIB.

11. Recode PSBs and assemble into IMS PSBLIB.
12. Code all DBD entries into IMSGEN source as DATABASE entries.
13. Recode CICS PDIR. Keep remote PSBs in the PDIR if users continue to access them as local DL/I databases in their remote CICS region. All other PSB entries should be copied into the IMSGEN source as APPLCTN entries.
14. Run IMS MODBLKS generation (plus security generation and copy jobs).
15. Run ACBGEN on OS/390.
16. Migrate DL/I databases from VSE to OS390.
17. Register databases with DBRC.
18. Code up and assemble IMS dynamic allocation modules for databases.
19. Migrate from COBOL/VS or COBOL II to COBOL/MVS. This involves minor rewrites and recompilation. See 8.8.4, "Migrate from COBOL/VS or COBOL II to COBOL/MVS" on page 121.
20. Migrate assembler application programs to ASMA90 assembler.
21. Change all programs handling dates and times to obtain them from the LE/MVS library routines in Year 2000 compliant format.
22. Change any programs accessing the UPSI byte bits 1 through 4. Consider recoding all affected batch programs to accept information using the INQY ENVIRONMENT command, through the APARM parameter of the IMSBATCH or DLIBATCH procedures.
23. Migrate application programs from the CICS macro-level to the CICS command-level API. This must include removing the DFHSEC macro from any customized sign-on programs, and deleting EXEC CICS ADDRESS CSA statement.
24. Rewrite CICS user exits that have CICS gateway macros and use the new CICS system programming interface (SPI) instead of the old control block DSECTS structure.
25. Migrate from local security to external security.

8.1.3 Postmigration Tasks

These should be done in parallel with system capacity testing or volume testing before you cut over to production:

- Install full external security using a security-server such as IBM RACF. RACF is the product explicitly described in this book, but other products could be used to achieve the same results.
- Recode batch CICS jobs as IMS BMPs.

8.1.4 Optional Migration Tasks

These are optional steps that can be performed in any order, or not at all. IBM recommends that you include them all in any conversion project:

- Migrate CICS DCT table to RDO.
- Convert terminal and program definitions to autoinstall. It can be done a bit at a time. Once autoinstall is activated, CICS continues to use a definition if it finds one, and creates one only if one does not already exist.

8.2 Introduction to SAFE II

The System for Advanced Financial Environment Interactive Interface (SAFE II) is an online financial communications system operating in a DOS/VSE, VSE/ESA environment under CICS/DOS/VS(CICS/VSE). Using VTAM/NCP or VTAM and with a data base interface to DOS/VS DL/I, SAFE II supplies sets of application programs for the host and the IBM 3600 Financial Communication System (FCS) or the IBM 4700 FCS controller to provide an online, multibranch banking environment.

SAFE II-supplied host application programs process savings and checking transactions for the controller and teller. The SAFE II application program in the controller provides basic SAFE II banking functions to branches even when communications with the host are disrupted or the host is otherwise unavailable.

SAFE II also provides a modified version of DATSMAP for the customer to access other CICS/VS 3270 application programs. In addition, SAFE II provides a set of macros as an aid for the customer to write his own SAFE II application programs. SAFE II customizing macros allow the customer to tailor SAFE II for a particular banking environment.

In summary, the functions of the SAFE II are these:

- To provide banking service to any customer at any online branch.
- To maintain complete cash control for each teller.
- For each bank customer, to maintain up-to-date records including:
  - Available balance.
  - All transactions processed during the day.
  - Status of each account.
  - Credit limit (checking).
  - Float credit limit (checking).
  - Checkbook information (checking).
- To maintain complete and up-to-date system audit information.
- To continue basic branch services when the central (host) computer is down or when communications to the host are disrupted.
- To aid the user in writing new SAFE II applications.
- To aid the user in customizing SAFE II for a particular banking environment.
- To aid the user in using the signature display PRPQ.

There is a large customer installation base in China. Many of these customers also have in-house added-on application programs based on SAFE II macros. We select SAFE II as our application migration sample because its large installation base could help many of the emerging-market country customers to speed up their upgrade processing.

8.3 Premigration Tasks

The following documents our project detail migration work.

8.3.1 Operating System and Subsystems

At this stage, OS/390 and all subsystems should already have been installed and application online regions built. Users should set security to off during this early migration period. Set all CICS JOURNALMODEL types to DUMMY (see Figure 20 on page 67) and include resource definitions in CICS and IMS for only those resources required to start the combined system.

The user may have performed the installation in-house and produced a system similar to that described in Chapter 3, “Introduction to IMS/ESA DBCTL” on page 19, or the company may have employed someone outside to do it.

8.3.2 Migrate CICS PCT and PPT to RDO on VSE

Omit this section if you have already migrated all online definitions to RDO.

The user should first define and initialize a CSD on VSE if it does not already have one. A sample job is shown at Figure 31 on page 85. The reason to do this is because CICS TS in OS/390 no longer provides DFHPPT and DFHPCT macros for users to do batch processing. The only solution is to migrate to VSE online CSD. Otherwise, users must define all the applications to CICS TS manually.

```
* $$  JOB JNM=CSDINIT,CLASS=B,.....

// JOB CSDINIT accounting-info
// DLBL CICCAT,'CICCAT',,VSAM
// DLBL OUTDD1,'IMS.SJCICSD.DFHCSD',,VSAM,CAT=CICCAT
// LIBDEF PHASE,SEARCH=(CICS.LOADLIB,CICS.TABLIB,PRD1.BASE,PRD2.BASE)
// EXEC IDCAMS
  DEFINE CLUSTER(NAME(CICS.DFHCSD)  -
                   VOLUMES(CIC001) -
                   KEYS(22 0) -
                   INDEXED -
                   RECORDS(1000 5000) -
                   RECORDSIZE(120 500) -
                   FREESPACE(10 10) -
                   SHAREOPTIONS(2)) -
                 DATA -
                   (NAME(CICS.DFHCSD.DATA)     -
                    CONTROLINTERVALSIZE(8192)) -
                 INDEX -
                   (NAME(CICS.DFHCSD.INDEX))     -
                 CATALOG(CICCAT)
/*
// EXEC DFHCSDUP
  VERIFY  FILE(DFHCSD)
  INITIALIZE
  UPGRADE USING(DFHCU17F)
  UPGRADE USING(DFHCU22F)
  UPGRADE USING(DFHCU23F)
  UPGRADE USING(DFHCU17I)
  UPGRADE USING(DFHCU22I)
  UPGRADE USING(DFHCU17R)
  LIST ALL OBJECTS
/*
/&
* $$ EOJ
```

Figure 31. Sample CICS/VSE JCL to Initialize CSD File

Then run the DFHCSDUP utility shown in Figure 32 on page 86 on the VSE machine to migrate all current PPT and PCT to online definitions. In the example, the suffixes are D0 for the PCT D1, and for PPT.

```
* $$  JOB JNM=MIGRATE,CLASS=B,...
// JOB MIGRATE accounting-info
// DLBL CICCAT,'CICCAT',,VSAM
// DLBL DFHCSD,'IMS.SJCICSD.DFHCSD',,VSAM,CAT=CICCAT
// LIBDEF PHASE,SEARCH=(CICS.LOADLIB,CICS.TABLIB,PRD1.BASE,PRD2.BASE)
// EXEC IDCAMS
  VERIFY FILE(DFHCSD)
/*
// EXEC DFHCSDUP
  MIGRATE TABLE(DFHPCTD0) TOGROUP(APPLPCT)
  MIGRATE TABLE(DFHPPTD1) TOGROUP(APPLPPT)
  COPY    GROUP(APPLPCT)  TO(APPLGRP)
  COPY    GROUP(APPLPPT)  TO(APPLGRP)
  DELETE  ALL GROUP(APPLPCT)
  DELETE  ALL GROUP(APPLPPT)
  ADD     GROUP(APPLGRP)  LIST(STARTLST)
  LIST    ALL OBJECTS
/*
/&
* $$ EOJ
```

Figure 32. Sample CICS/VSE JCL to Migrate PPT and PCT to RDO

8.4 Migration Tasks

A long list of tasks need to be done for the migration. We, therefore, divide them into groups with related functions, described in 8.6, "CICS Migration Tasks" on page 87 through 8.8, "Application Migration Tasks" on page 113.

8.5 Migration Environment

It is ideal if both VSE and OS/390 can access the same DASDs. Then, the system can make all VSE files, databases, and libraries available to OS/390. To do this, you must *export* the VSE master catalog and *import* it to OS/390 as one of the OS/390 user-catalogs. All VSE application and system environment files can then be accessed by OS/390. In case of incompatible files, you can *punch* (a VSE special terminology; it means write) data to a temporary sequence file and make it immediately available to OS/390.

The most likely environment is that in which both VSE and OS/390 share access to a few work DASDs that they use as the staging area. Files that need to migrate to OS/390 are moved or converted and punched to the staging disks, to be immediately picked up by the OS/390 side. This saves a lot of data transport time. However, you need to carefully coordinate work on both sides. Files generated in one side may not be recognized by the other and could be overwritten by accident. Permanent files should never be placed on these shared DASDs until the migration is completed.

The least effective migration method is to unload all VSE data to tape and place them on another tape driver to be read by OS/390. The current ITSO configuration does not allow our OS/390 sysplex to share DASDs with the VM system. As a result, we had to use the ineffective way to transport our data.

8.6 CICS Migration Tasks

The following sections deal with the CICS TS aspect of migration tasks.

8.6.1 Copy the CICS CSD from CICS/VSE to CICS TS

Since the CSD is normally on a disk that is online to VSE (probably SYSWK1), you will not be able to make it available to MVS to migrate it directly. Therefore you must proceed as follows:

1. Copy the CSD on the VSE image to tape.
2. Copy the tape CSD into a temporary CSD on MVS.
3. Copy all application groups from the temporary CSD to run-time CSD, CICS.SJCICSD.DFHCSD.
4. Update the CICS start-up parameters to include the new resources.

8.6.1.1 Step 1. Copy the CSD on VSE to Tape

This is must be done using a utility available on both VSE and MVS. We use DITTO to transfer the file here, although we could equally well have used IDCAMS REPRO. The tape is assumed to be labeled as TAP001 and Figure 33 shows our run-time JCL.

```
* $$  JOB JNM=CSDDUMP,CLASS=B,...

// JOB CSDDUMP accounting-info
// PAUSE   Pls. mount tape on 3A0
// DLBL CICCAT,'CICCAT',,VSAM
// DLBL DFHCSD,'IMS.SJCICSD.DFHCSD',,VSAM,CAT=CICCAT
// TLBL TAPECSD,'CICS.DFHCSD.TAPECOPY',,'TAP001'
// ASSGN SYS002,320
// LIBDEF PHASE,SEARCH=(CICS.LOADLIB,CICS.TABLIB,PRD1.BASE,PRD2.BASE)
// EXEC IDCAMS
   VERIFY FILE(DFHCSD)
/*
// EXEC DITTO
 $$DITTO VT INPUT=DFHCSD,OUTPUT=TAPECSD
/*
/&
* $$ EOJ
```

Figure 33. Sample CICS/VSE JCL to Copy CSD to Tape

8.6.1.2 Step 2. Copy the Tape to Temporary CSD on OS/390

Mount the tape produced in Step 1 on OS/390. To run these jobs as shown in Figure 34 on page 88, you must have already defined and initialized the CICS CSD as described in Chapter 7, "System Environment Customization" on page 57.

```
//CSDTEMP JOB (999,POK),'Copy CSD from tape',CLASS=A,MSGCLASS=T
//*
//CSDALLOC EXEC IDCAMS,REGION=4M
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
   DEFINE CLUSTER(NAME(CICS.TEMP.DFHCSD) -
                  MODEL(CICS.SJCICSD.DFHCSD))
/*
//*
//CSDCOPY  EXEC PGM=DITTO,REGION=6M
//TMPCSD    DD  DISP=OLD,DSN=CICS.TEMP.DFHCSD
//TAPECSD   DD  DISP=OLD,DSN=CICS.DFHCSD.TAPECOPY,
//          UNIT=TAPE,VOL=SER=TAP001,LABEL=(,SL)
//SYSIN     DD  *
 $$DITTO TV INPUT=TAPECSD,OUTPUT=TMPCSD
/*
//
```

Figure 34. Sample JCL to Copy CSD from VSE Tape

8.6.1.3 Step 3. Copy from Temporary CSD to Run-Time CSD

Once this job has completed, merge the definitions from this CSD into the run-time CSD on MVS. A job like that shown in Figure 35 should be used.

```
//CSDTEMP JOB (999,POK),'Copy CSD from tape',CLASS=A,MSGCLASS=T
//*
//CSDMERGE EXEC PGM=DFHCSDUP
//STEPLIB   DD  DISP=SHR,DSN=CICS.SJICSD.SDFHLOAD
//DFHCSD    DD  DISP=SHR,DSN=CICS.SJCICSD.DFHCSD
//TMPCSD    DD  DISP=SHR,DSN=CICS.TEMP.DFHCSD
//SYSIN     DD  *
  COPY GROUP(APPLGRP) TO(APPLGRP) FROMCSD(TMPCSD)
  ADD  GROUP(APPLGRP) TO LIST(APPLLST)
  VERIFY
  LIST ALL
/*
//
```

Figure 35. Sample JCL to Copy CSD to Run-Time CSD

8.6.1.4 Step 4. Update CICS Start-up Parameters

The CICS start-up parameter overrides must be updated. At this point, there is a record in CICS.SJCICSD.PARMLIB(OVERRIDE) which looks like:

```
GRPLIST=(DFHLIST,TEMPDLST)
```

Change this to

```
GRPLIST=(APPLLST,DFHLIST,TEMPDLST)
```

This allows you to pick up all application definitions in the future as the user adds them into CICS. If users have CICS up at the time, then they could install the resources with the command

```
CEDA INSTALL GROUP(APPLGRP)
```

8.6.2 Copy All CICS Tables

Once you have PCT and PPT resources in the CSD, the next step is to copy the other CICS tables (the source version) to OS/390. You must then process most of these tables on OS/390. The steps involved are shown below:

1. Make sure you have a data set on MVS called TABSRC (or whatever, according to your installation's naming convention).
2. Write (punch for VSE) the source of the tables to tape files.
3. Copy the tape files to MVS flat files, and then to PDS members.

8.6.2.1 Step 1. Allocate TABSRC on OS/390 Using ISPF 3.2

If this data set does not already exist, allocate CICS.SJCICSD.TABSRC, using ISPF 3.2 as shown in Figure 36. Use the following parameters in Figure 36.

```
Data Set Name  .. . : CICS.SJCICSD.TABSRC

Space units . . . . . CYLS
Primary quantity. . . 10
Secondary quantity. . 2
Directory blocks. . . 30
Record format . . . . FB
Record length . . . . 80
Block size  . . . . . 3120
```

Figure 36. Allocation Parameters for TABSRC Data Set

8.6.2.2 Step 2. Punch the Source of the Tables to Tape Files

Transfer most of the CICS tables from a VSE source library, which may be either in a native VSE library, or a VSE ICCF library. Both options are shown below.

The job shown in Figure 37 on page 90 writes (punches) out a series of table source members from the VSE library. The members are named DFHxxxD0.A, where xxx is the table type, such as FCT or DCT, D0 is the table suffix, and .A is the member type. Change the names to suit your installation's naming convention. Create dummy members for those tables you have no source for, with at least one comment line in each.

```
* $$  JOB JNM=TABDUMP,CLASS=B,...

// JOB TABDUMP accounting-info
// PAUSE  Pls. mount tape on 3A0
// MTC REW,3A0
// TLBL SYSPCH,'CICS.TEMP.T.DFHTCTD0',,'TAP002',1,1
// ASSGN SYSPCH,3A0
// EXEC LIBR
  ACCESS SUBLIB=CICS.TABLIB
  PUNCH DFHTCTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHFCTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHDCTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHPLTPI.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHPLTSD.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHTSTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHXLTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHMCTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHCLTD0.A FORMAT=IEBUPDTE EOF=NO
  PUNCH DFHTLTD0.A FORMAT=IEBUPDTE EOF=YES  *the last PUNCH must have
                                             EOF=YES
/*
// MTC REW,3A0
// ASSGN SYSPCH,UA
/&
* $$ EOJ
```

Figure 37. Transfer CICS VSE Tables Source Library to Tape

For ICCF, there are several ways to transfer members to an OS/390 PDS. Here is the one we used and the following are the procedures:

- Define a temporary VSE library to catalog ICCF members as VSE library members.
- Run transfer job to catalog ICCF members to VSE library, then write to the tape as input to OS/390 IEBUPDTE utility.
- Define PDS data set on OS/390 system.
- Run OS/390 IEBUPDTE utility to load tape as PDS members (see Section 8.6.2.4, "Step 4. Loading the Tables Source to MVS" on page 92).

The first two procedures are shown in Figure 38 on page 91 as one VSE job. The job writes out a series of table source members from an ICCF library, called 24 (LIB=24). The ICCF members are to be called xxxD0, where xxx is the table type, such as FCT or DCT, and D0 is the table suffix. Defining the PDS data set on OS/390 is shown as 8.6.2.1, "Step 1. Allocate TABSRC on OS/390 Using ISPF 3.2" on page 89.

```
* $$  JOB JNM=TABDUMP,CLASS=B,...

// JOB TABDUMP accounting-info
// TLBL SYSPCH,'CICS.TEMP.T.DFHTCTD0',,'TAP002',1,1
// ASSGN SYSPCH,3A0
// LIBDEF PHASE,SEARCH=(PRD1.BASE,PRD2.BASE)
// EXEC LIBR
  ACCESS SUBLIB=TEMPUNCH.LIB
  CATALOG DFHTCTD0.A
* $$ SLI ICCF=TCTD0,LIB=24
/+
  PUNCH DFHTCTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHFCTD0.A
* $$ SLI ICCF=FCTD0,LIB=24
/+
  PUNCH DFHFCTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHDCTD0.A
* $$ SLI ICCF=DCTD0,LIB=24
/+
  PUNCH DFHDCTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHPLTPI.A
* $$ SLI ICCF=PLTPI,LIB=24
/+
  PUNCH DFHPLTPI.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHPLTSD.A
* $$ SLI ICCF=PLTSD,LIB=24
/+
  PUNCH DFHPLTSD.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHTSTD0.A
* $$ SLI ICCF=TSTD0,LIB=24
/+
  PUNCH DFHTSTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHXLTD0.A
* $$ SLI ICCF=XLTD0,LIB=24
/+
  PUNCH DFHXLTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHMCTD0.A
* $$ SLI ICCF=MCTD0,LIB=24
/+
  PUNCH DFHMCTD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHCLTD0.A
* $$ SLI ICCF=CLTD0,LIB=24
/+
  PUNCH DFHPDRD0.A FORMAT=IEBUPDTE EOF=NO
  CATALOG DFHTLTD0.A
* $$ SLI ICCF=TLTD0,LIB=24
/+
  PUNCH DFHTLTD0.A FORMAT=IEBUPDTE EOF=YES   * The last punch must has
                                             EOF=YES
/*
// MTC RUN,SYSPCH
// ASSGN SYSPCH,UA
/&
* $$ EOJ
```

Figure 38. Sample JCL to Write Table Source from VSE ICCF Library to Tape

8.6.2.3 Step 3. Copy from Temporary CSD to Run-Time CSD

Once this job has completed, merge the definitions from this CSD into the run-time CSD on MVS. A job like that shown in Figure 35 on page 88 should be used.

8.6.2.4 Step 4. Loading the Tables Source to MVS

This job uses the tape created in Step 2 as input and uses it to create members in the library CICS.SJCICSD.TABSRC on OS/390, as shown in Figure 39.

```
//ICCFISPF JOB ICCFISPF,'PGMRNAME',
// CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1),
// NOTIFY=IMSVSE2,
// REGION=32M
//*
//STEP1    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   DSNAME=VSE.VSELIB.DDA,DISP=OLD
//SYSIN    DD   DSNAME=PUNCH.ICCF.OS390,DISP=OLD,VOL=SER=TAP002,
//         LABEL=(2,BLP),DCB=BLKSIZE=80,UNIT=3490
/*
```

Figure 39. Sample JCL to Load the Table Source from VSE Tape to PDS

Note

VSE LIBR PUNCH function does not provide HDR2 information on the standard MVS tape label. You must code DCB=BLKSIZE=80 on input tape file DD statement (SYSIN in Figure 39) and use the BLP parameter to bypass MVS label processing.

8.6.3 Copy All Current DL/I DBDs and PSBs

Following the procedures of 8.6.2, "Copy All CICS Tables" on page 89, we unload all application DBDs and PSBs to tape and move them over to the OS/390 system. Jobs and coding should be identical with those shown in Figure 36 on page 89 through Figure 39 and are not repeated here. The steps are these:.

1. Allocate DBDSLIB and PSBSLIB libraries.

   Allocate IMS.SJIMSD.DBDSLIB and IMS.SJIMSD.PSBSLIB if they do not yet exist. Apply the same method described in 8.5, "Migration Environment" on page 86 but replacing data set name and disk space allocation parameter.

2. Transfer (punch) the tables source code to tape.

   Use jobs similar to the example in Figure 37 on page 90 or Figure 38 on page 91 to unload all tables source to tape. Replace:

   ```
        PUNCH DFHTCTDO.A FORMAT=IEBUPDTE EOF=NO
   ```

   with

   ```
        PUNCH <DBDname>.A FORMAT=IEBUPDTE EOF=NO
   or
        PUNCH <PSBname>.A FORMAT=IEBUPDTE EOF=NO
   ```

3. Use the OS/390 IEBUPDTE utility to load the source tape into the MVS PDS library.

8.6.4 Migrate CICS Tables TCT and FCT to RDO on OS/390

CICS TS no longer supports the batch resource definition for FCT and VTAM TCT terminals. The best way is to run MIGRATE function of the CICS TS resource definition utility to migrate the table into a CSD group and add to CICS start-up resource list called *GRPLIST*. However, BDAM files, TCAM or sequential terminals are still not online-installable, so they must remain as batch utility jobs.

8.6.4.1 Words about Migration

Figure 40 on page 94 is a JCL procedure supplied by IBM for the purpose of compiling and link-editing of any general CICS table macro definitions. This procedure must be used when specifying MIGRATE=YES on DFHxxx TYPE=INITIAL macro.

For FCT, insert this:

```
DFHFCT TYPE=FILE,GROUP=<group> before file definition
```

The procedure creates a separate load module for migration (see Figure 41 on page 97).

For TCT, the assembly of a TCT leads to the production of a single text file. However, two members are created when the file is link-edited into the load library. The members are:

- DFHTCTxx, which contains the non-RDO-eligible definitions in control block format.
- DFHRDTxx, which contains the RDO-eligible definitions in command format.

Be aware of the existence of these two tables if you have to copy or move assembled TCT tables between load libraries.

The procedure is not stored in the standard procedure library, SDFHPROC. We comment out the SMP/E step because of time constraints. We recommend that you comment it out during the early migration stage, even though you must reinstate it later to get the right system maintenance.

```
//DFHAUPLE PROC ASMBLR=ASMA90,
//       LNKED=IEWL,
//       RENTATT=NORENT,
//       INDEX='CICSTS12.CICS',
//       INDEX2='CICS.SJCICSD',
//       OUTC='*',
//       REG=4M,
//       NAME=SDFHLOAD,
//       WORK=SYSDA,
//*      SMPPGM=SMPE,
//*      GZONE='&GZONECSI',
//       ZNAME=&TNAME,
//       OPTIONS=&SMPEOPTIONS
//*
//*      THIS PROCEDURE IS USED
//*      TO GENERATE CICS/ESA TABLES.
//*
//*      THE FOLLOWING JCL SHOULD BE USED
//*      TO EXECUTE THIS PROCEDURE
//*
//*      //TABLEASM EXEC DFHAUPLE
//*      //ASSEM.SYSUT1 DD *
//*         .
//*         . TABLE GENERATION MACROS
//*         .
//*      /*
//*
//ASSEM  EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=&OUTC
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=&&TEMPPDS(MACROS),UNIT=&WORK,DISP=(,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
//             SPACE=(400,(800,200,10))
//ASM    EXEC PGM=&ASMBLR,
//        PARM='SYSPARM(INITIAL),DECK,NOOBJECT,ALIGN',
//        REGION=&REG,COND=(3,LT,ASSEM)
//SYSLIB   DD  DSN=&INDEX..SDFHMAC,DISP=SHR
//         DD  DSN=&INDEX..SDFHSAMP,DISP=SHR
//         DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SYS1.MODGEN,DISP=SHR
//*
//SYSUT1   DD  UNIT=&WORK,SPACE=(1700,(400,400))
//SYSUT2   DD  UNIT=&WORK,SPACE=(1700,(400,400))
//SYSUT3   DD  UNIT=&WORK,SPACE=(1700,(400,400))
//SYSPUNCH DD  DSN=&&OBJMOD,DISP=(,PASS),UNIT=&WORK,
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
//             SPACE=(400,(400,100))
//SYSPRINT DD  SYSOUT=&OUTC
//SYSIN    DD  DSN=&&TEMPPDS(MACROS),
//             VOL=REF=*.ASSEM.SYSUT2,
//             DISP=(OLD,PASS)
```

Figure 40 (Part 1 of 3). IBM-Supplied JCL to Assemble and Load Macro Definitions

```
//BLDMBR EXEC PGM=IEBUPDTE,PARM=NEW,
//        COND=((3,LT,ASSEM),(7,LT,ASM))
//SYSPRINT DD  DUMMY
//SYSUT2   DD  DSN=&&TEMPPDS,
//             VOL=REF=*.ASSEM.SYSUT2,
//             DISP=(OLD,PASS)
//SYSIN    DD  DSN=&&OBJMOD,DISP=(OLD,DELETE)
//LNKEDT EXEC PGM=&LNKED,
//        PARM='&RENTATT,LIST,XREF,LET,NCAL',
//        COND=((3,LT,ASSEM),(7,LT,ASM),
//        (3,LT,BLDMBR))
//SYSUT1   DD  UNIT=&WORK,SPACE=(1024,(100,50))
//SYSPRINT DD  SYSOUT=&OUTC
//SYSLMOD  DD  DSN=&INDEX2..&NAME,DISP=SHR
//SYSPUNCH DD  DSN=&&TEMPPDS,
//             VOL=REF=*.ASSEM.SYSUT2,
//             DISP=(OLD,PASS)
//SYSLIN   DD  DSN=&&TEMPPDS(LNKCTL),
//             VOL=REF=*.ASSEM.SYSUT2,
//             DISP=(OLD,PASS)
//ZNAME    EXEC PGM=IKJEFT01,REGION=&REG,DYNAMNBR=99,
// PARM='%DFHZNSET ZNAME(&ZNAME) OPTIONS(&OPTIONS)'
//SYSPROC  DD DISP=SHR,DSN=&INDEX..SDFHMAC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//OUTFILE  DD DISP=(NEW,PASS),DSN=&&SETBDY,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
// UNIT=&WORK,SPACE=(400,(1,1))
//*
//*SMP   EXEC PGM=&SMPPGM,REGION=&REG,
//*       COND=((3,LT,ASSEM),(7,LT,ASM),
//*       (3,LT,BLDMBR),(7,LT,LNKEDT))
//*SMPOUT  DD  SYSOUT=&OUTC
//*SYSPRINT DD SYSOUT=&OUTC
//*SMPCSI  DD  DSN=&GZONE,DISP=SHR
//*SMPCNTL DD  DSN=&&SETBDY,DISP=(OLD,DELETE)
//*        DD  DSN=&&TEMPPDS(SMPCNTL),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
//*SMPJCLIN DD DSN=&&TEMPPDS(SMPJCL1),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
//*        DD  DSN=&&TEMPPDS(MACROS),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
//*        DD  DSN=&&TEMPPDS(SMPJCL2),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
```

Figure 40 (Part 2 of 3). IBM-Supplied JCL to Assemble and Load Macro Definitions

```
//*        DD  DSN=&&TEMPPDS(LNKCTL),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
//*        DD  DSN=&&TEMPPDS(SMPEOF),
//*            VOL=REF=*.ASSEM.SYSUT2,
//*            DISP=(OLD,PASS)
//*
//DELTEMP EXEC PGM=IEFBR14
//TEMP     DD  DSN=&&TEMPPDS,
//             VOL=REF=*.ASSEM.SYSUT2,
//             DISP=(OLD,DELETE)
/*
```

Figure 40 (Part 3 of 3). IBM-Supplied JCL to Assemble and Load Macro Definitions

8.6.4.2 FCT Migration

The FCT source should be changed so that it can be assembled and migrated to RDO:

- Using the ISPF editor, change the FCT source (CICS.SJCICSD.TABSRC(DFHFCTD0) for our project).

  Locate statement of: DFHFCT TYPE=INITIAL, then,

  - Optionally change the suffix to SUFFIX=MG.
  - Add the subparameter MIGRATE=YES.

- Add this statement:

  ```
  DFHFCT TYPE=GROUP,GROUP=APPLFCT
  ```

  at some logical point so that related files can be treat as a resource group.

- Remove any DLI entries, and replace with

  ```
  DFHFCT TYPE=DATASET,ACCMETH=VSAM
  ```

  entries that describe DL/I databases.

The modified FCT source should look like Figure 41 on page 97.

```
*
FCTDA    DFHFCT TYPE=INITIAL,STARTER=YES,SUFFIX=MG,MIGRATE=YES
*        DFHFCT TYPE=DATASET,            Delete or comment DL/I related
*              DATASET=CUSTLOG,          definitions
*              ACCMETH=DL/I,
*              OPEN=INITIAL
***********************************************************************
***            ENTRIES FOR SAFE II FILES                           ***
***********************************************************************
         DFHFCT TYPE=GROUP,GROUP=VSEFCT
FFFFJRN  DFHFCT TYPE=DATASET,         --- user files  ------------     *
               DATASET=FFFFJRN,                                        *
               ACCMETH=VSAM,                                           *
               SERVREQ=(UPDATE,BROWSE),                                *
               FILSTAT=(ENABLED,OPENED),                               *
               LSRPOOL=NONE,                                           *
               LOG=NO,                                                 *
               RECFORM=(FIXED),                                        *
               BUFND=5,            NUMBER OF DATA BUFFERS              *
               STRNO=4             NUMBER OF CONCURRENT REQUESTS
      .......
```

Figure 41. Modified FCT Source for Migration to RDO

Then assemble and migrate the source to RDO using the JCL shown below:

```
//FCTLOAD  JOB .....
//*
//    JCLLIB ORDER=VSE.ICCF.LIB
//*
//STEP1 EXEC PROC=DFHAUPLE
//ASSEM.SYSUT1 DD DISP=SHR,DSN=VSE.ICCF.LIB(DDAFCTC1)
/*
```

Use the JCL shown in Figure 42 to merge the newly converted FCT group into a CICS start-up group list.

```
//IMSVSE2E  JOB ACCNT#,IMSVSE2,
//             NOTIFY=IMSVSE2,
//             MSGLEVEL=(1,1)
//*
//STEP1   EXEC PGM=DFHCSDUP,REGION=20M
//STEPLIB   DD DSNAME=CICS.SJCICSD.SDFHLOAD,DISP=SHR
//          DD DSNAME=CICS.SJCICSD.TABLOAD,DISP=SHR
//DFHCSD    DD DSNAME=CICS.SJCICSD.DFHCSD,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  MIGRATE TABLE(DFHFCTDA)
  ADD     GROUP(VSEFCT) LIST(APPLLIST)
  VERIFY
/*
```

Figure 42. Sample JCL to Merge Newly Converted Table into a CSD Goup

Once this has been done, if there are any entries in the FCT of the type ACCMETH=DAM, then:

- Change ACCMETH=DAM to ACCMETH=BDAM.
- Change the TYPE=INITIAL to SUFFIX=D0,MIGRATE=COMPLETE.
- Change the SIT override member CICS.SJCICSD.PARMLIB(OVERRIDES) so that the line FCT=NO becomes FCT=D0.

Then assemble into the TABLOAD library as shown in Figure 43.

```
//FCTASMD JOB (accounting-info),'Assemble FCT',CLASS=A,MSGCLASS=T
//*
//FCTASM EXEC PROC=TABASM
//C.SYSIN   DD DISP=SHR,DSN=CICS.SJCICSD.PARMLIB(FCTD0)
//L.SYSLMOD DD DISP=SHR,DSN=CICS.SJCICSD.TABLOAD
//L.SYSLIN  DD
//          DD *
        NAME DFHFCTD0(R)
/*
//
```

Figure 43. Sample JCL to Assemble FCT

BDAM files are not supported by the online definition feature.

8.6.4.3 TCT Migration

The TCT source should be changed so that it can be assembled and migrated to RDO. Make these changes:

- Using the ISPF editor, change the TCT source in CICS.SJCICSD.TABSRC(DFHTCTD0) so that on the DFHTCT TYPE=INITIAL macro, change the suffix to SUFFIX=MG, and add the subparameter MIGRATE=YES. The result is shown in Figure 44 on page 99.
- You still need to use DFHAUPLE procedure to assemble the TCT source.

```
TCTC1    DFHTCT TYPE=INITIAL,STARTER=YES,SUFFIX=C1,                    X
               ACCMETH=VTAM,                                           X
               MIGRATE=YES
*
         DFHTCT TYPE=GROUP,GROUP=VESCONN    <====
         DFHTCT TYPE=SYSTEM,                                           *
               ACCMETH=VTAM,                                           *
               SYSIDNT=HQAC,                                           *
               NETNAME=CICSHQAC,                                       *
               TRMTYPE=LUTYPE62,                                       *
         DFHTCT TYPE=MODESET,                                          *
               SYSIDNT=HQAC,                                           *
               MODENAM=ROSEISC,                                        *
               MAXSESS=(20,10),                                        *
               BUFFER=2048,           MAX DATA LENGTH TO SEND          X
               RUSIZE=2048
               ......

         DFHTCT TYPE=GROUP,GROUP=VSEPRT     <====
L86P     DFHTCT TYPE=TERMINAL,        VTAM LOCAL 3286 PRINTER          X
               TCTUAL=255,                                             X
               TRMIDNT=L86P,                                           X
               NETNAME=P42L308,       DESCRIBED IN VTAM B.BOOK         X
               ACCMETH=VTAM,                                           X
               TRMTYPE=L3284,                                          X
               TRMMODL=2,                                              X
               TIOAL=2200,                                             X
               TRMSTAT=RECEIVE,                                        X
               RELREQ=(YES,YES),                                       X
               FEATURE=(PRINT),                                        X
               PGESIZE=(24,80),                                        X
               PGESTAT=AUTOPAGE
               .....
         DFHTCT TYPE=GROUP,GROUP=VSETERM
D040     DFHTCT TYPE=TERMINAL,                                         *
               ACCMETH=VTAM,                                           *
               FEATURE=(AUDALARM,UCTRAN),                              *
               GMMSG=YES,                                              *
               NETNAME=D04001,                                         *
               PGESTAT=PAGE,                                           *
               RELREQ=(,YES),                                          *
               TCTUAL=255,                                             *
               TIOAL=300,                                              *
               TRMIDNT=D040,                                           *
               TRMMODL=2,                                              *
               TRMSTAT=TRANSCEIVE,                                     *
               TRMTYPE=L3277
               .....
```

Figure 44. Sample JCL to Migrate Modified TCT Source to RDO

- Add DFHTCT TYPE=GROUP entries to order the terminals into logical groupings. It is a good idea to put all the consoles into a group of their own. For example, when we add groups called CONSOLS, TCTGRP1, TCTGRP2, TCTGRP3, TCTGRP4, TCTGRP5, the JCL is then as shown in Figure 45 on page 100.

```
//TCTMIGR  EXEC PGM=DFHCSDUP
//STEPLIB  DD DISP=SHR,DSN=CICS.SJICSD.SDFHLOAD
//         DD DISP=SHR,DSN=CICS.SJICSD.TABLOAD
//DFHCSD   DD DISP=SHR,DSN=CICS.SJCICSD.DFHCSD
//SYSIN    DD *
  MIGRATE TABLE(DFHTCTMG)
  ADD GROUP(CONSOLS) LIST(APPLLST)
  ADD GROUP(TCTGRP1) LIST(APPLLST)
  ADD GROUP(TCTGRP2) LIST(APPLLST)
  ADD GROUP(TCTGRP3) LIST(APPLLST)
  ADD GROUP(TCTGRP4) LIST(APPLLST)
  ADD GROUP(TCTGRP5) LIST(APPLLST)
  VERIFY
  LIST ALL
/*
//
```

Figure 45. Sample JCL to Load TCT Definitions into CSD

Once this has been done, if you specified ACCMETH=(VTAM,NONVTAM) or defaulted the ACCMETH, and you have any TCAM, or sequential devices, you must then:

- Change the TYPE=INITIAL to SUFFIX=D0,MIGRATE=COMPLETE.
- Change the SIT override member CICS.SJCICSD.PARMLIB(OVERRIDES) so that the line TCT=NO becomes TCT=D0.
- Then assemble into the TABLOAD library with the JCL shown in Figure 46.

```
//FCTRDO  JOB (accounting-info),'Assemble TCT',CLASS=A,MSGCLASS=T
//FCTASM  EXEC PROC=TABASM
//C.SYSIN   DD DISP=SHR,DSN=CICS.SJCICSD.PARMLIB(TCTD0)
//L.SYSLMOD DD DISP=SHR,DSN=CICS.SJCICSD.TABLOAD
//L.SYSLIN  DD
//          DD *
        NAME DFHTCTD0(R)
/*
//
```

Figure 46. Sample JCL to Assemble the TCT

Note

You cannot use BSC terminals with CICS TS. If you have any of these devices, you must convert them in VTAM to VTAM/BSC. That is, use VTAM = LBUILD/LOCAL definitions for locally attached devices, and LINE/CLUSTER/TERMINAL definitions for devices attached through an NCP. They are then defined to CICS using RDO as a 3270 TYPETERM.

8.6.5 Recode the SRT

We strongly advise you to use the supplied default SRT. If you have coded your own for VSE, you will have to change it, because even where abend codes are similar, OS/390 issues 3 digit abend codes (such as 0C4, B37) whereas VSE issues 2 digit codes (20,0F). For many system codes, no exact equivalent exists.

If you are using either the supplied program DFHSRTRR or your own programs that use only standard CICS SPI functions, then you probably don't need to change the SRT beyond finding the equivalent abend codes. However, if your SRT program is issuing other commands or macros, then you need to rewrite the routines completely. The details of how to code this type of SRT program are beyond the scope of this book. See:

- *CICS for VSE/ESA 2.3 Resource Definition (Macro), SC33-0709*
- *VSE Messages and Codes, Volume 1, SC33-6507*

for the description of VSE abend codes and how CICS handles them, and see:

- *CICS Transaction Server for OS/390 CICS Resource Definition Guide, SC33-1684*
- *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683*
- *OS/390 MVS System Codes, GC28-1780*

for how to write the new SRT program.

8.6.6 Assemble and Link Other CICS Tables

A few other CICS tables need to be migrated at this time. They are described here.

8.6.6.1 DCT

The CICS DCT can either be assembled on OS/390 or migrated to the CSD. We advise that you simply assemble it into TABLOAD first, then migrate it later under the optional tasks.

8.6.6.2 Other Tables

You need to reassemble and link any tables CLT, MCT, PLTSD, TLT, TST and XLT tables into a TABLOAD library at this point. Take these steps for every such table present:

- Edit the source in CICS.SJCICS.TABSRC to change the SUFFIX parameter of the TYPE=INITIAL to SUFFIX=D0.
- Edit the SIT overrides member CICS.SJCICSD.PARMLIB(OVERRIDES) to change xxT=NO to xxT=D0.
- Assemble using a job similar to that in Figure 46 on page 100.

8.6.7 Change CICS DASD Logs to MVS Logger DASD Log Streams

7.5, “CICS TS Journaling and Logging” on page 63 gives a complete description of CICS log streams. What we describe here is the change-over method for the migration project only.

All DL/I logging is now done by IMS DBCTL. If you are using the CICS logs only for recovery of the DL/I database in the VSE environment, you do not need any CICS logging in the new OS/390 CICS IMS/DBCTL environment. IMS knows — independently of CICS — whether a transaction succeeded or not. However, for reference, you need to do the following to define the logging:

1. Size the logs

   You may spend some time optimizing this step, as described in the installation guide, or you can use the maximum parameters for the first CICS system and tune it as you add applications. To do this well, you must create a copy of a VSE CICS log on OS/390, and use it as input to the DFHLSCU utility.

2. Optionally, define the journal models to OS/390.

   Use code like that shown in Figure 47.

```
//DEFLOGS  JOB ...
//LOGDEFN  EXEC PGM=IXCMIAPU
//STEPLIB  DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*******************************************************************
//*
//* Define DASD-only model log streams for CICS system log.
//*
//* The LOWOFFLOAD and STG_SIZE values are for illustration
//* only -- substitute values appropriate for your environment.
//*
//*******************************************************************
//SYSIN    DD *
  DATA TYPE(LOGR) REPORT(NO)
  DEFINE LOGSTREAM NAME(SC62.DFHLOG.MODEL)
                   MODEL(YES)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(3000)
                   LOWOFFLOAD(60) HIGHOFFLOAD(95)
  DEFINE LOGSTREAM NAME(SC62.DFHSHUNT.MODEL)
                   MODEL(YES)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(500)
                   LOWOFFLOAD(0) HIGHOFFLOAD(80)
  DEFINE LOGSTREAM NAME(SC62.USERJRNL.MODEL)
                   MODEL(YES)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(500)
                   LOWOFFLOAD(0) HIGHOFFLOAD(80)
  DEFINE LOGSTREAM NAME(SC62.GENERAL.MODEL)
                   MODEL(YES)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(500)
                   LOWOFFLOAD(0) HIGHOFFLOAD(80)
/*
//
```

Figure 47. Sample JCL to Define Journal Models to OS/390

3. Run the job shown in Figure 48 to define CICS online region DASD-only log stream.

```
//DEFLOGS  JOB ...
//LOGDEFN  EXEC PGM=IXCMIAPU
//STEPLIB  DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****************************************************************
//*
//* Define DASD-only log streams for CICS system log.
//*
//* The LOWOFFLOAD and STG_SIZE values are for illustration
//* only -- substitute values appropriate for your environment.
//*
//*****************************************************************
//SYSIN    DD *
  DATA TYPE(LOGR) REPORT(NO)
  DEFINE LOGSTREAM NAME(CICS.SJCICSD.DFHLOG)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(3000)
                   LOWOFFLOAD(60) HIGHOFFLOAD(95)
  DEFINE LOGSTREAM NAME(CICS.SJCICSD.DFHSHUNT)
                   DASDONLY(YES)
                   MAXBUFSIZE(64000)
                   STG_SIZE(500)
                   LOWOFFLOAD(0) HIGHOFFLOAD(80)
/*
//
```

Figure 48. Sample JCL to Define Logstream to OS/390

4. Then define them to RACF. Commands like these will do:

```
RDEF logstrm CICS.** OWNER(CICS)     UACC(READ)
PE   CICS.** CLASS(LOGSTRM) id(CICS) ACCESS(ALTER)
PE   CICS.** CLASS(LOGSTRM) id(IMS)  ACCESS(ALTER)
```

5. Then map them to CICS and install the definitions.

   Map the MVS log stream definitions for CICS log to use a batch job with CICS utility DFHCSDUP (see Figure 49 on page 104). It will enable CICS logging at the next start, and can be removed and changed easily if definitions are not satisfactory:

```
//DEFJ     EXEC PGM=DFHCSDUP,REGION=4M
//STEPLIB   DD  DSN=CICS.SJCICSD.SDFHLOAD,DISP=SHR
//DFHCSD    DD  DSN=CICS.SJCICSD.DFHCSD,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
  DEFINE JOURNALMODEL(DFHLOG) JOURNALNAME(DFHLOG)     TYPE(MVS)
           GROUP(LOG62) STREAM(CICS.SJCICSD.DFHLOG)
  DEFINE JOURNALMODEL(DFHSHUNT) JOURNALNAME(DFHSHUNT) TYPE(MVS)
           GROUP(LOG62) STREAM(CICS.SJCICSD.DFHSHUNT)
  ADD  GROUP(LOG62) LIST(LIST62)
/*
//
```

Figure 49. Sample JCL to Map the JOURNALMODEL to CICS Log to OS/390

You then need to amend the CICS start-up overrides in CICS.SJCICSD.PARMLIB(OVERRIDE) to change the GRPLIST statement. Use this:

```
GRPLIST=(DFHLIST,TEMPDLST,LIST62)
```

You could also use RDO to define this procedure and install it immediately.

6. Start CICS START=INITIAL the first time. Shut it down, and then start again with START=COLD. Subsequent starts can be START=AUTO.

8.7 IMS Migration Tasks

8.4, “Migration Tasks” on page 86 deals primarily with the CICS TS application online region. The following tasks are mainly for the IMS DBCTL application region.

8.7.1 Recode DBDs and Assemble into IMS DBDLIB

The following are step by step procedure we have used and tested.

8.7.1.1 Change Source

You must change the source of the DBDs that have been installed in IMS.SJIMSD.DBDSLIB to make them run under OS/390. Note the following points for an OS/390 DBD definition:

- FIELD statements must have the START and BYTES operands coded.
- ACCESS statements need to be coded into the DBD according to database file type. See 8.7.1.2, “DBD ACCESS Statement” on page 105 for more information.
- If the database consists of HIDAM VSAM files, then the database file is ESDS and the related index file is KSDS. Once you generate DL/I DBD in IMS DBCTL, there are some changes in related VSAM cluster definitions. So check the DBD generation output carefully, and use the recommended VSAM definition parameters to define new VSAM files on OS/390. Do not use old VSE VSAM definitions, as they may cause database errors. Figure 50 on page 105 is a sample DBD output with VSAM definition recommendation.

```
      ......

Stmt   Source Statement                                   HLASM R2.0  1998
 54+*,* * * * * * * * * * * * * * * * * * * * * * * * * * *
 55+*,*
 56+*,   RECOMMENDED VSAM DEFINE CLUSTER PARAMETERS
 57+*,*
 58+*,* * * * * * * * * * * * * * * * * * * * * * * * * * *
 60+*,* * * * * * * * * * * * * * * * * * * * * * * * * * *
 61+*,*
 62+*,*                       *NOTE2
 63+*,*  DEFINE CLUSTER (NAME(FFFBTLE) NONINDEXED -
 64+*,*         RECORDSIZE (2041,2041) -
 65+*,*         CONTROLINTERVALSIZE (2048))
 66+*,*
 67+*,* *NOTE2 - SHOULD SPECIFY DSNAME FOR DD FFFBTLE
 68+*,*
 69+*,* * * * * * * * * * * * * * * * * * * * * * * * * * *
 71+         DC    AL2(2053)          FREE SPACE VALUE           @BI5ZZ
 72+         DC    AL1(0)             SPACE SEARCH ALGORITHM #   @BIL8W
 73+         DC    AL1(0)             RESERVED                   @BI5ZZ
 74+         DC    F'-1'              END OF PREFIX

 >>>> DATA BASE DEFINITION FOR  FFFBTLE <<<<
    ......
```

Figure 50. DBD Generation Output with Recommended VSAM Definition

- In the VSE DL/I environment, ESDS file record size = (CIsize - 10). The file record size must be (CIsize - 7) in OS/390.
- FBA devices are not supported by MVS. As a result, you need to remove the DEVICE= parameter of the DATABASE statement.

8.7.1.2 DBD ACCESS Statement

You need to code the ACCESS statement according to the database type. Table 9 and Table 10 on page 106 and Table 11 on page 106 summarize the parameters.

Table 9. ACCESS Statement Parameters for HDAM DBD

VSE ACCESS parameter	**OS/390 DBD**	**OS/390 DBD parameter**	**Remarks**
RMRTN=xxxx	DBD	RMNAME=(xxxx,	*mod*
CIANPT=nnnn	DBD	RMNAME=(,nnnn	*anch*
PRIMCI=nnnn	DBD	RMNAME=(,,nnnn	*rbn*
RILIM=nnnn	DBD	RMNAME=(,,,nnnn)	*bytes*
SEGM=nnnn	SEGM	PARENT=0	root segment
SEQFLD=nn	SEGM	NAME=(,SEQ,)	root segment
SEQVAL=c	SEGM	NAME=(,SEQ,c)	*U or M*

Table 10. ACCESS Statement Parameters for HIDAM Primary Index

VSE ACCESS parameter	OS/390 DBD	OS/390 DBD parameter	Remarks
REF=	DBD	ACCESS=INDEX	Needs a separate DBD coded
SEGM=nnnn	LCHILD •••	NAME=nnnn	in index DBD
SEQFLD=xxxx	LCHILD •••	INDEX=xxxx	in index DBD

••• In addition an LCHILD statement must be coded in the data DBD.

Table 11. DBD Parameters Replacing ACCESS Statement for HD Secondary Index

VSE ACCESS parameter	OS/390 DBD	OS/390 DBD parameter	Remarks
REF=	DBD	ACCESS=INDEX	Needs a separate DBD coded
SEGM=target	LCHILD & XDFLD positioning	after SEGM statement	*See explanation below*
SEQSEQ=source	XDFLD	SEGMENT=source	
SEQFLD=xxxx	XDFLD	SRCH=xxxx	
SEQVAL=xxxx	XDFLD	SUBSEQ=xxxx	
SUPVAL=xxxx	XDFLD	NULLVAL=xxxx	
SUPRTN=xxxx	XDFLD	EXTRTN=xxxx	

The non-ACCESS statement approach consists of an LCHILD and XDFLF following the target segment SEGM statement, in the data portion of the database. Associated FIELD statements are needed in data DBD if /sx or /ck are used. An index DBD defining the index database with its LCHILD statement referencing the target segment in the database is also required.

8.7.1.3 Run DBDGEN

DBDGEN procedure is generated during IMS full generation processing. Run DBDGEN as shown in the following statement:

```
//DBFSAMD3 EXEC DBDGEN,MBR=dbdname,SOUT='*'
//C.SYSIN  DD DISP=SHR,
//         DSN=IMS.SJIMSD.DBDSLIB(dbdname)
//*
```

8.7.2 Recode PSBs and Assemble into IMS PSBLIB

The procedures are as follows.

8.7.2.1 Change Source

PSBs fortunately require much less work than DBDs. However, the user should check the source in IMS.SJIMSD.PSBSLIB to ensure the following:

- PSB LANG= must be COBOL, PL/I, PL/1, PASCAL, ASSEM or blank if using OS/390 Language Environment. No other values, or spellings, or trailing blanks are accepted.
- START= must be coded in the SENFLD statement.
- Check that none of the following functions remain:
  - Virtual fields
  - Field types Z,E,D,L
  - Automatic data conversion
  - Field exit routines.

  If any of these are found, you need to recode the programs to remove them.

8.7.2.2 Run PSBGEN

PSBGEN is a standard procedure created during IMS full-generation processing. Run a PSBGEN as follows:

```
//DBFSAMD3 EXEC PSBGEN,MBR=psbnsme,SOUT='*'
//C.SYSIN  DD DISP=SHR,
//         DSN=IMS.SJIMSD.PSBSLIB(psbname)
//*
```

8.7.3 Run ACBGEN

Once you have completed DBDGEN and PSBGEN, you must rerun the ACBGEN utility and the online copy. The ACBGEN procedure is shown in Figure 138 on page 259. The job to call this is shown in Figure 51.

```
* JCL in IMS.SJIMSD.DBDC.JCL(ACBGEN)

//  JCLLIB ORDER=IMS.SJIMSD.PROCLIB
//    EXEC ACBGEN,COMP=POSTCOMP
//SYSIN DD *
    BUILD PSB=psbname
//
```

Figure 51. Sample ACBGEN JCL

Once ACBGEN is completed, then run the online copy job shown in Figure 52 on page 108.

```
*  JCL in IMS.SJIMSD.DBDC.JCL(ACBCOPY)

//IMSDGOLC JOB (B3562,W941),'IMS TE-DBDC',
//         MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//*
//      JCLLIB ORDER=IMS.SJIMSD.PROCLIB
//***************************************************************
//* IMS ONLINE CHANGE UTILITY
//***************************************************************
//*
//OLC1    EXEC OLCUTL,IMSID=SJIMSD,IN=S,OUT=U,TYPE=ACBLIB
```

Figure 52. Sample JCL for ACB Online Copy

Once this is complete, issue the online change commands to IMS:

```
<IMSD>MOD PREPARE ACBLIB.
<IMSD>MOD COMMIT.
```

If IMS is not running, then run the job IMS.SJIMSD.DBDC.JCL(ACBCOPAB) as shown in Figure 137 on page 257

8.7.4 Code DBD Entries in IMSGEN

Once you have migrated to CICS TS and IMS DBCTL, there is no more DLZACT module, and DL/I database files can no longer be defined in the CICS FCT. Programs are either defined in IMS DBCTL or they remain in CICS TS. All DBD entries must be converted into DATABASE entries of IMS Stage 1 generation source. You can do this by editing the source (in CICS.SJCICSD.TABSRC(DFHDDRD0) for our case) and:

- Removing the TYPE=INITIAL macro
- Removing the TYPE=FINAL macro
- Removing the END instruction
- Changing 'DFHDLDIR TYPE=ENTRY,' to 'DATABASE RESIDENT '

Then append the contents to member IMS.SJIMSD.STAGE1.SOURCE(APPLDB), to form a complete IMS Stage 1 generation source.

There is also a set of migration aid macros to convert local DL/I PSB and DBD into IMS generation macros. See F.3, "CICS Local DL/I to IMSGEN Conversion Aid" on page 284 for information.

8.7.5 Recode CICS PDIR

Keep remote PSBs in the PDIR if users will continue to access them as local DL/I databases in their remote CICS regions. All other PSBs entries should be copied into the IMSGEN source and converted into APPLCTN entries. Users can do this by editing source in CICS.SJCICSD.TABSRC(DFHPDRD0) thus:

- Create a copy of the member.
- On the copy,
  - Remove the TYPE=INITIAL macro
  - Remove the TYPE=FINAL macro

- Remove the END instruction
- Remove ′DFHDLPSB TYPE=ENTRY,′ entries with SYSIDNT coded
- Change ′DFHDLPSB TYPE=ENTRY,′ to ′APPLICATION RESIDENT, PGMTYPE=BATCH,SCHDTYP=PARALLEL ′
- Add the LANG= parameter for all COBOL or PL1 PSBs

You can also use the migration aid macros to convert local DL/I PSB and DBD into IMS generation macros. See F.3, "CICS Local DL/I to IMSGEN Conversion Aid" on page 284 for information.

Then append the contents to member IMS.SJIMSD.STAGE1.SOURCE(APPLDB). You next need to remove from CICS.SJCICSD.TABSRC(DFHPDRD0) all entries not coded with SYSIDNT=.

If any entries are left, then this source should be assembled using a job similar to that in Figure 46 on page 100 and the SIT overrides member CICS.SJCICSD.PARMLIB(OVERRIDES) should be changed from PDIR=NO to PDIR=D0.

8.7.6 Run IMS MODBLKS Generation

Run the IMS MODBLKS generation, and the security generation and the online copy job.

In the IMS DBCTL environment, with the exception of installing system maintenance, the only type of generation users will ever run is a MODBLKS generation. The process for this is like running a full generation (see B.8, "The IMS DBCTL Full Generation Processing" on page 250), except that the user does not run the SMP steps. Also, users can install it while IMS is running instead of needing a cold start. The MODBLKS generation is a three-step process. Figure 53 on page 110 and Figure 54 on page 111 show sample JCL for two of the three steps.

```
* JCL in IMS.SJIMSD.DBDC.JCL($1MODBKG)

//IMSGENS1 JOB (999,POK),
// CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1),
// NOTIFY=IMSVSE1,
// REGION=32M
//*
//         SET SYS1=SJIMSD
//         SET SYSSMP=IMS610
//*
//*********************************************************************
//* INSTALL/IVP IMS 6.1
//*
//* SKELETON: DFSIXSC2
//*
//* FUNCTION: IMS SYSTEM DEFINITION PREPROCESSOR - FOR NAME VALIDATION
//*********************************************************************
//*
//PREPROC  EXEC PGM=DFSPRE00,PARM='N,N',TIME=(600)
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//SYSSUMPR DD SYSOUT=*
//SYSCYLPR DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IMS.&SYS1..STAGE1.SOURCE
//         DD DISP=SHR,DSN=&SYSSMP..GENLIB
//         DD DISP=SHR,DSN=&SYSSMP..GENLIBA
//         DD DISP=SHR,DSN=&SYSSMP..GENLIBB
//SYSIN80  DD DISP=(NEW,DELETE),DSN=&&SYSIN,
//            UNIT=3390,SPACE=(CYL,(5,5)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DSORG=PS)
//SYSUT1   DD UNIT=3390,SPACE=(CYL,(5,5))
//SYSLIN   DD DISP=(NEW,DELETE),DSN=&&SYSLIN,
//            UNIT=3390,SPACE=(CYL,(1,5)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DSORG=PS)
//SYSPRT80 DD DISP=(NEW,DELETE),DSN=&&SYSPRT,
//            UNIT=3390,SPACE=(CYL,(5,5)),
//            DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3146,DSORG=PS)
//SYSCOBJ  DD DUMMY,DCB=BLKSIZE=80
//SYSIN    DD DISP=SHR,
//            DSN=IMS.SJIMSD.STAGE1.SOURCE(MBLKGEN)
```

Figure 53. Sample JCL to Scan MODBLKS Stage 1 Source

```
* JCL in IMS.SJIMSD.DBDC.JCL($2MODBKG)

//IMSGENS1 JOB (999,POK),
// CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1),
// NOTIFY=IMSVSE1,
// REGION=32M
//*
//         SET SYS1=SJIMSD
//         SET SYSSMP=IMS610
//*
//STAGE1   EXEC PGM=ASMA90,PARM='NOOBJ,DECK',TIME=(600)
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IMS.&SYS1..STAGE1.SOURCE
//         DD DISP=SHR,DSN=&SYSSMP..GENLIB
//         DD DISP=SHR,DSN=&SYSSMP..GENLIBA
//         DD DISP=SHR,DSN=&SYSSMP..GENLIBB
//SYSPUNCH DD DISP=SHR,
//            DSN=IMS.SJIMSD.STAGE1.SOURCE(STAGE2)
//SYSUT1   DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSUT2   DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSUT3   DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSIN    DD DISP=SHR,
//            DSN=IMS.SJIMSD.STAGE1.SOURCE(MBLKGEN)
```

Figure 54. Sample JCL for MODBLKS Stage 1 Generation

In most cases, the MODBLKS generation source is identical with the FULL generation source, except for the application system and the following.

Macro: Use

```
IMSCTRL  SYSTEM=(VS/2,(MODBLKS,DBCTL),390)
```

instead of

```
IMSCTRL  SYSTEM=(VS/2,(ALL,DBCTL),390)
```

and use

```
IMSGEN NODE=(IMS.SJIMSD,IMS.SJIMSD,IMS610)
```

instead of

```
IMSGEN NODE=(IMS.SJIMSD,IMS610,IMS610)
```

Once this is complete, submit and run the Stage 2 jobs from member IMS.SJIMSD.STAGE1.SOURCE(STAGE2), which is the output of Stage 1.

This should be followed by these other procedures:

1. Run the Security generation. See Figure 136 on page 258 for security generation JCL procedure and Figure 55 on page 112 for the execution JCL.
2. The JCL to run the online copy is shown in Figure 56 on page 112.

```
*  JCL in IMS.SJIMSD.DBDC.JCL($4SECGEN)

//IMSDGSEC JOB (999,POK),'IMS TE-DBDC',CLASS=A,
//         TIME=(,30),MSGLEVEL=(1,1),MSGCLASS=T
//*
//***************************
//*  IMS SECURITY GEN
//***************************
//SECURITY EXEC AIGSEC,
//            RESLIB='IMS.SJIMSD.RESLIB',
//            MATRIX='IMS.SJIMSD.MATRIX',
//            MODBLKS='IMS.SJIMSD.MODBLKS',
//            RSUF=D
//S.SYSIN DD *
)( AGN BMP01
   AGPSB ALL
/*
```

Figure 55. Sample JCL for IMS Security Generation

```
* JCL in IMS.SJIMSD.DBDC.JCL($5ONLCHN)

//IMSDGOLC JOB (999,POK),'IMS TE-DBDC',
//          MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//*
//       JCLLIB ORDER=IMS.SJIMSD.PROCLIB
//*******************************************************
//* IMS ONLINE CHANGE UTILITY
//*******************************************************
//OLC1    EXEC OLCUTL,IMSID=SJIMSD,IN=S,OUT=U,TYPE=MATRIX
//OLC2    EXEC OLCUTL,IMSID=SJIMSD,IN=S,OUT=U,TYPE=MODBLKS
//
```

Figure 56. Sample JCL to Run the Online Copy

Install with these IMS online change commands

```
/MOD PREPARE MODBLKS
/MOD COMMIT
```

while IMS is running. If IMS is down, an off-line change job is also available. Refer to Figure 137 on page 257 for information and warm-start IMS to activate it afterward.

8.8 Application Migration Tasks

The following subsections deal with the application aspect of migration tasks.

8.8.1 Migrate DL/I Databases from VSE to OS/390

The simplest migration can occur if the databases are defined as VSAM files or files on a CKD disk on VSE, or on a disk with no non-VSAM files. In that case, the disk can either be physically or logically moved, and the VSE VSAM catalog imported to the OS/390 master catalog (as a user-catalog). Once it is installed, it can then be converted to ICF and all the necessary alias entries can subsequently be defined. However, most migrations are more complicated. The database types you are likely to have are these:

- HISAM
- SHISAM (HISAM with root segments only)
- HDAM
- HIDAM — as a VSAM file
- Index

Unless you can connect the VSE disk to OS/390, you should move the nondatabase files using IDCAMS EXPORT/IMPORT, and move the database files by unloading them as DL/1 databases, and reloading them using IMS utilities. A similar process should be used also for any non-DL/I files.

8.8.1.1 Define Alias in OS/390

Define the high-level qualifiers of the data set names to RACF and the catalog. We are defining the high level qualifier VSE and the user catalog in which we are defining it is:

```
CATALOG.TOTICF1.USERCAT.
```

Use TSO Option 6 to issue the following commands:

```
DEF ALIAS(NAME('VSE') rel('CATALOG.TOTICF1.USERCAT'))
AG  VSE          OWNER(SYS1) SUPGROUP(SYS1)
AD  'VSE.**'     GENERIC  OWNER(IMSVSE02) UACC(UPDATE)
AD  'VSE.DB.**'  GENERIC  OWNER(IMSVSE02) UACC(UPDATE)
AD  'VSE.FIL.**' GENERIC  OWNER(IMSVSE02) UACC(UPDATE)
```

8.8.1.2 Exporting VSAM Files from VSE

There are several ways to port VSE VSAM files to OS/390 VSAM. We give only the EXPORT/IMPORT example here. Users can EXPORT VSE/VSAM files and IMPORT directly to an OS/390 ICF catalog. Figure 57 on page 114 is an example.

```
* $$  JOB JNM=EXPORTF,CLASS=B,...

// JOB EXPORTF accounting-info
// PAUSE    Pls. check the tape
// MTC REW,3A0
// DLBL VSESPUC,'VSE.USER.CATALOG',,VSAM
// DLBL VSEFILE1,'VSE.FILE.ONE',,VSAM,CAT=VSESPUC
// DLBL VSEFILE2,'VSE.FILE.TWO',,VSAM,CAT=VSESPUC
// DLBL VSEFILE3,'VSE.FILE.THREE',,VSAM,CAT=VSESPUC
// TLBL FEXPT1,'IMS.FIL.FILE1.EXP',,'TAPDB1',1,1
// TLBL FEXPT2,'IMS.FIL.FILE2.EXP',,'TAPDB1',1,2
// TLBL FEXPT3,'IMS.FIL.FILE2.EXP',,'TAPDB1',1,3
// ASSGN SYS005,3A0
// LIBDEF PHASE,SEARCH=(PRD1.BASE,PRD2.BASE)
// EXEC IDCAMS
      EXPORT VSE.FILE.ONE OUTFILE(FEXPT1  -
      ENV(BLKSZ(32760) NOREWIND PDEV(2400))    -
      RECM TEMP NINHS NOERASE
      EXPORT VSE.FILE.TWO OUTFILE(FEXPT2  -
      ENV(BLKSZ(32760) NOREWIND PDEV(2400))    -
      RECM TEMP NINHS NOERASE
      EXPORT USER1.FIL.FILE1 OUTFILE(FEXPT3  -
      ENV(BLKSZ(32760) NOREWIND PDEV(2400))    -
      RECM TEMP NINHS NOERASE
/*
// MTC RUN,3A0
/&
* $$ EOF
```

Figure 57. Sample JCL to Export VSE VSAM Files

8.8.1.3 Import Files to MVS

The exported files can then be imported into the OS/390 environment using the JCL shown in Figure 58.

```
//IMPORDB JOB (999,POK),'IMS TE-DBDC',
//        MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//*
//IMPORT  EXEC PGM=IDCAMS,REGION=6M
//SYSPRINT DD SYSOUT=*
//FEXPT1   DD DISP=OLD,DSN=IMS.FIL.FILE1.EXP,
//         UNIT=3490,VOL=SER=TAPDB1,LABEL=(1,SL)
//FEXPT2   DD DISP=OLD,DSN=IMS.FIL.FILE2.EXP,
//         UNIT=3490,VOL=SER=TAPDB1,LABEL=(2,SL)
//FEXPT3   DD DISP=OLD,DSN=IMS.FIL.FILE3.EXP,
//         UNIT=3490,VOL=SER=TAPDB1,LABEL=(3,SL)
//SYSIN    DD *
 IMPORT OUTDATASET(VSE.FIL.FILE1) INFILE(FEXPT1) INTOEMPTY
 IMPORT OUTDATASET(VSE.FIL.FILE2) INFILE(FEXPT2) INTOEMPTY
 IMPORT OUTDATASET(VSE.FIL.FILE3) INFILE(FEXPT3) INTOEMPTY
/*
//
```

Figure 58. Sample JCL to Import VSAM Files to OS/390

8.8.1.4 Unloading DL/I Databases from VSE

We also use the UNLOAD/RELOAD utilities, DLZRRC00 and DFSRRC00, to port databases across. Before porting, make sure DL/I DBD has already been generated and the DL/I VSAM files have been defined based on recommended parameters in the DBD generation output list. Only one file can be moved per job. See the sample JCL in Figure 59.

```
* $$  JOB JNM=EXPORDB,CLASS=B,...

// JOB UNLOADDB    Unload DB under VSE
// PAUSE    Pls. mount the tape
// MTC REW,3A0
// DLBL DB1CAT,'DB1CAT',,VSAM
// DLBL DBD1,'USER1.DB.FILE1',,VSAM,CAT=DB1CAT
// DLBL DBD1I,'USER1.DB.INDEX1',,VSAM,CAT=DB1CAT    * If DB has index
// ASSGN SYS010,IGN      * No restart *
// ASSGN SYS011,3A0      * First copy *
// ASSGN SYS012,IGN      * No second copy *
// TLBL HDUNLD1,'IMS.DB.FILE1.EXP',,'TAPDB2'
// LIBDEF PHASE,SEARCH=(VSE.DBDLIB)
// EXEC DLZRRC00,SIZE=300K
ULU,DLZURGU0,DB1
REW=R
CHKPT=NO
/*
// MTC REW,3A0
/&
* $$ EOF
```

Figure 59. Sample JCL to Unload VSE DL/I Database

8.8.1.5 Reload Databases to OS/390

The OS/390 side of unload is IMS reload utility shown in Figure 60.

```
//IMPORDB JOB (999,POK),'IMS TE-DBDC',
//        MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//*
//IMPORT  EXEC PGM=DFSRRC00,PARM='ULU,DFSURGL0,DBD1,,,,,,,,,,,N',
//        REGION=20M
//STEPLIB  DD  DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD  DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//DFSRESLB DD  DSN=IMS.SJIMSD.RESLIB,DISP=SHR
//IMS      DD  DISP=SHR,DSN=IMS.SJIMSD.DBDLIB
//SYSPRINT DD  SYSOUT=*
//DFSUINPT DD  DISP=OLD,DSN=IMS.DB.FILE1.EXP,
//         UNIT=3490,VOL=SER=TAPDB2,LABEL=(1,SL)
//DBD1     DD  DISP=SHR,DSN=USER1.DB.FILE1
//DBD1I    DD  DISP=SHR,DSN=USER1.DB.INDEX1
//DFSURCDS DD  DSN=IMSESA.RLCDS,DISP=(OLD,KEEP),
//         UNIT=SYSDA,VOL=SER=TOTTSD
//DFSVSAMP DD  *
8192,20
20480,20
VSRBF=8192,20
VSRBF=4096,5
VSRBF=2048,5
VSRBF=512,5
IOBF=(2048,5)
IOBF=(4096,5)
IOBF=(8192,10)
//DFSCTL   DD *
SBPARM ACTIVE=NO
/*
```

Figure 60. Sample JCL to Reload Databases on OS/390

8.8.1.6 Creating Dynamic Allocations

In order to access the database files conveniently, the next step is to code and assemble dynamic allocation macros for them. The source then compiles and links them into the APF-authorized library, called *USERLIB*. A sample job is shown below, and the JCL for dynamic allocation is shown in Figure 128 on page 246.

```
This source would be placed in member IMS.SJIMSD.DALSRC(DFSIVD1)

        DFSMDA TYPE=INITIAL
        DFSMDA TYPE=DATABASE,DBNAME=IVPDB1     Add as many of these
        DFSMDA TYPE=DATASET,DDNAME=DFSIVD1,    in the same member as  x
             DSNAME=IMS.SJIMSD.DFSIVD1,        you want               x
             DISP=SHR
        DFSMDA TYPE=FINAL
        END

The JCL to assemble it will be found in IMS.SJIMSD.JOBS(DYNALLOC)

//*
//       SET SYS1='SJIMSD.'
//*
//MAKE   EXEC IMSDALOC,SYS1=&SYS1
//SYSIN  DD DISP=SHR,DSN=IMS.&SYS1.DALSRC(DFSIVD1)     **
```

Figure 61. Sample Source and JCL to Assemble Database Dynamic Allocations

8.8.2 Register Databases with DBRC

The sample JCL to register databases to IMS DBCTL is shown in Figure 62.

```
* JCL in IMS.SJIMSD.DBDC.JCL(DBRCIDB)

//*
//         SET SYS1='SJIMSD.'
//*
//DBRC     EXEC PGM=DSPURX00,REGION=3M
//STEPLIB   DD  DSN=IMS.&SYS1..USERLIB,DISP=SHR
//          DD  DSN=IMS.&SYS1..RESLIB,DISP=SHR
//PROCLIB   DD  DSN=IMS.&SYS1..PROCLIB,DISP=SHR
//JCLOUT    DD  SYSOUT=(A,INTRDR)
//JCLPDS    DD  DSN=IMS.&SYS1..JCLLIB,DISP=SHR
//IMS       DD  DISP=SHR,DSN=IMS.&SYS1..DBDLIB
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
  INIT.DB   DBD(CZFBTBE)
  INIT.DBDS DBD(CZFBTBE) DDN(CZFBTBE) DSN(CASE.ZJLS.CZFFTBE) GENMAX(4)
  INIT.DB   DBD(CZFBTBK)
  INIT.DBDS DBD(CZFBTBK) DDN(CZFBTBK) DSN(CASE.ZJLS.CZFFTBK) GENMAX(4)
      .............
      (as many of these as needed)
//
```

Figure 62. Sample JCL to Register Databases to IMS DBCTL

8.8.3 Migrate VSE DL/I Batch to IMS Batch

The DL/I batch API is compatible with both VSE DL/I and IMS DBCTL. However, DL/I batch applications need to be converted if they were written in an earlier COBOL version. Also, the following modifications are needed in the VSE DL/I batch program and in the JCL:

- Users must declare pointers in the COBOL LINKAGE SECTION for PSB entries passed by a DL/I batch utility.
- Users must put the USING pcb-pointer parameter in PROCEDURE DIVISION.
- Set PCB working storage to the address of the pcb-pointer.

Figure 63 on page 119 shows part of the DL/I batch sample program (to set pcb-pointer) and Figure 64 on page 121 shows the execution JCL for reference.

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.  COBLCU.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT    SECTION.
 FILE-CONTROL.
       SELECT INFILE     ASSIGN TO CUST
                         ORGANIZATION IS SEQUENTIAL
                         ACCESS MODE IS SEQUENTIAL
                         FILE STATUS IS FORSTAT.
*
 DATA DIVISION.
 FILE            SECTION.
                 BLOCK CONTAINS 0 RECORDS.
 01  INREC.
     10  REC-KEY         PIC X(10).
     10  DATA1           PIC X(169).
     10  FILLER          PIC X(02).
     10  DATA2           PIC X(500).
*
 WORKING-STORAGE SECTION.

 77  ISRT                PIC  X(04)       VALUE "ISRT".
 77  F-SYNC              PIC  X(04)       VALUE "SYNC".

 01  OUT-SEG.
     05  OUTREC          PIC  X(679).

 01  OUT-AREA.
     05  OUT-KEY         PIC  X(10).
     05  OUT-DATA1       PIC  X(169).
     05  OUT-DATA2       PIC  X(500).
     ........

 LINKAGE SECTION.

 01  DBPCB1 POINTER.      1  PCB pointer

******************************************************************
*I/O PCB
******************************************************************

 01  LTERMPCB.
     05  LOGTTERM        PIC  X(08).
     05  FILLER          PIC  X(02).
     05  TPSTAT          PIC  X(02).
     05  IODATE          PIC  X(04).
     05  IOTIME          PIC  X(04).
     05  FILLER          PIC  X(02).
     05  SEQNUM          PIC  X(02).
     05  MOD             PIC  X(08).
```

Figure 63 (Part 1 of 2). DL/I Batch Program Sample

```
*****************************************************************
*DATA BASE PCB
*****************************************************************

 01  DBPCB.
     05  DBDNAME         PIC  X(08).
     05  SEGLEVEL        PIC  X(02).
     05  FILLER          PIC  9(08) COMP.
     05  SEGNAMFB        PIC  X(08).
     05  LENKEY          PIC  9(08) COMP.
     05  SENSSSEGS       PIC  9(08) COMP.
     05  KEYFB           PIC  X(20).
     05  FILLER REDEFINES KEYFB.
         07  KEYFB1      PIC  X(10).
         07  FILLER      PIC  X(10).

*****************************************************************
 PROCEDURE DIVISION USING DBPCB1.         2  USING parameter
*****************************************************************
*
 SET ADDRESS OF DBPCB TO ADDRESS OF DBPCB1.      3  set pointer

 ACCEPT WARVAL FROM SYSIN.
 COMPUTE REC-MAX = REC-MAX * NUM-DIST * NUM-CUST.
 OPEN INPUT INFILE.

 PERFORM WITH TEST BEFORE VARYING REC-CNT
         FROM 1 BY 1 UNTIL REC-CNT > REC-MAX
   READ INFILE AT END GO TO NORMAL-END
   END-READ
   MOVE REC-KEY TO OUT-KEY
   MOVE DATA1   TO OUT-DATA1
   MOVE DATA2   TO OUT-DATA2
   MOVE OUT-AREA TO OUTREC
   CALL "CEETDLI"
     USING ISRT, DBPCB, OUT-SEG, OUT-SSA
   PERFORM CHKSTAT THRU CHKSTAT-EXIT
 END-PERFORM.

 ........

 TERM.
   DISPLAY DISP-NUMREC UPON CONSOLE.
   CLOSE INFILE.
   STOP RUN.
```

Figure 63 (Part 2 of 2). DL/I Batch Program Sample

```
    User COBOL DL/I batch load model-------+      +----PSB name
                                           ↓      ↓
//STEP01    EXEC  PGM=DFSRRC00,PARM='DLI,LOADER2,CCFPTLL,,,,,,,,,,,N',
//          REGION=32M
//DFSRESLB    DD  DSN=IMS.SJIMSD.RESLIB,DISP=SHR
//*           DD  DSN=VSE.LOAD.DDA,DISP=SHR
//STEPLIB     DD  DSN=IMS.SJIMSD.RESLIB,DISP=SHR
//            DD  DSN=VSE.LOAD.DDA,DISP=SHR
//IMS         DD  DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//            DD  DSN=IMS.SJIMSD.PSBLIB,DISP=SHR
//SYSPRINT    DD  SYSOUT=*
//DFSVSAMP    DD  *
8192,20
20480,20
VSRBF=8192,20
VSRBF=4096,5
VSRBF=2048,5
VSRBF=512,5
IOBF=(2048,5)
IOBF=(4096,5)
IOBF=(8192,10)
//SYSIN       DD  *
1 00100100119
2 000 000000000100000 000000720 000000000010000 00103500131 1
/*
```

Figure 64. Sample JCL to Run COBOL DL/I Batch

8.8.4 Migrate from COBOL/VS or COBOL II to COBOL/MVS

This will involve minor rewrites and recompilation. See Chapter 5, "Application Environment" on page 33 for the changes needed to move COBOL programs across.

As an alternative, you can buy and run the IBM product CCCA (Product 5785-CCC) against the COBOL source on VSE. See Appendix F, "Conversion Aid" on page 281, for more information.

8.8.5 Migrate Application Programs to ASMA90 Assembler

ASMA90, the high level assembler, is stricter than all of its predecessors in flagging for errors. Some sloppy coding practices that earlier compilers flagged as MNOTEs, are now flagged as Warnings and thus can cause assemblies to fail where they ran successfully before.

A common example is that if a comma was omitted in coding a MACRO call, prior assemblers would often simply set the remaining parameters to default values. With ASMA90, these would now be flagged as syntax errors.

8.8.5.1 CICS Application Programs

CICS programs written using the standard CICS API only do not need to be rewritten. Exits need to be recoded as part of the CICS installation, where differences in CICS release cause the functionality to need to be changed. Users do not need to change any of the DL/I calls.

See *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683* for more information.

A CICS AUTOINSTALL exit should work without modification.

8.8.5.2 CICS Application Programs with Nonstandard Features

CICS programs using any nonstandard features, macro-level calls, undocumented features, or direct addressing of CICS control blocks must be rewritten. Any CICS program making direct calls to the operating system routines should be discarded in any case. Database exits should need recoding only if they access operating system facilities.

8.8.5.3 Batch Programs

The DL/I calls do not need recoding to work. However, there are many other possible reasons for recoding. Each program should be looked at individually. Some reasons for recoding would be:

- If a program intercepts an abend code from VSE, then it would need recoding to trap the equivalent MVS abend code (if there is one). Note that many codes describe different conditions. Users need to reexamine many of these completely. See

  - *OS/390 MVS System Codes, GC28-1780*
  - *VSE Messages and Codes, Volume 1, SC33-6507*

  .

  for more information.

- If a program uses system macros or calls an SVC directly, then the program needs recoding. See

  - *CICS for VSE/ESA 2.3 System Programming Reference, SC33-0711*
  - *OS/390 MVS Authorized Assembler Services Guide, GC28-1763*

  for more information.

- If a program calls a POWER macro, then this would need to be rewritten to use equivalent JES facilities. See

  - *VSE/POWER Application Programming Guide, SC33-6574*
  - *OS/390 JES2 Macros, SC28-1795*

  for more information. JES2 does not have such a callable interface.

8.8.6 Rewrite Programs Accessing the VSE UPSI Byte

The function of the UPSI byte is replaced by the use of the PARM=′ ′ on job step EXEC statement in MVS. The UPSI still can be used in the current version of COBOL for MVS and VM, but we recommend recoding programs that access the UPSI byte to accept it from a PARM=parameter instead. The following is an example of coding to accept access from an UPSI setting in OS/390 in a COBOL program:

```
     UPSI-0 IS TYPE-SWITCH
        ON-STATUS IS QTR-END
        OFF-STATUS IS MTH-END

The JCL EXEC statement would be
//STEP75 EXEC PGM=PROCPGM,PARM=(/UPSI(10000000))
```

Users may also prefer to recode affected batch programs to accept information using the INQY ENVIRONMENT command, by way of the APARM parameter of the IMSBATCH or DLIBATCH procedures.

8.8.7 Rewrite All JCL

There is no simple algorithm to change JCL. Note that in MVS JCL, the EXEC card comes before the data sets that the program uses. In VSE JCL, the EXEC card comes afterward, but before its instream data. Table 12 attempts to show some equivalence of JCL card and function from VSE to OS/390. This is not an exact correspondence, partly because of the different catalog structures and device handling on MVS, and the very different structure of the two sets of JCL statements.

Table 12 (Page 1 of 2). JCL and Job Functions — OS/390 Equivalents of VSE

VSE & POWER JCL	**OS/390 MVS & JES2 JCL**
* $$ CTL	none
* $$ DATA	none
* $$ EOJ	//
* $$ FLS	MVS Cancel command
* $$ JOB	// JOB and /*JOBPARM statements
* $$ JOB DISP=K	MVS started task
* $$ LST	// DD SYSOUT= and /*OUTPUT and // OUTPUT
* $$ PUN	// DD SYSOUT= and /*OUTPUT and // OUTPUT
* $$ RDR	none. This function is not supported.
* $$ SLI	// INCLUDE or // DD (concatenated DD s)
* $$ SLI and $$ DATA	// INCLUDE but only into JCL
// ASSGN	// DD UNIT=... parameters.
// CLOSE	Function done by program.
// DATE	Operator SET DATE= command
// DLBL	// DD
// EXEC	// EXEC PGM=
// EXEC PROC=	// EXEC PROC= or // EXEC . PROCs must contain an // EXEC card.
// EXTENT	// DD SPACE=, VOL=SER= parameters
// IF and ON and GOTO	// IF and // THEN and // ELSE and // ENDIF
// ID	// JOB USER=, PASSWORD= parameters
// JCLEXIT	// none
// JOB	// JOB or presence of an // EXEC card.
// LIBDEF PHASE and LIBDROP PHASE	//STEPLIB DD concatenation
// LIBDEF and LIBDROP other than PHASE	//SYSLIB DD concatenation
// LIBLIST	none
// LIBSERV	none
// LISTIO	none
// MTC	none

Table 12 (Page 2 of 2). JCL and Job Functions — OS/390 Equivalents of VSE	
VSE & POWER JCL	**OS/390 MVS & JES2 JCL**
// OPTION	none
// PAUSE	Use TSO commands instead.
// PROC	// PROC Note that MVS PROCs must contain an EXEC card.
// PWR	none
// QUERY	Use operator D commands
// RESET	none
// RSTRT	// JOB RESTART= parameter
// SETPARM	// SET
// SETPRT	// OUTPUT or // DD SYSOUT=
// STDOPT	none
// SYSDEF	none
// TLBL	// DD UNIT=3480 or some other tape device
// UPSI	// EXEC PARM=' ' parameter
// VDISK	none. Use VIO or temporary datasets instead (// DD with no DSN= or // DD DSN=&&...)
// ZONE	Operator SET TIME= command
instream data	//SYSIN DD * followed by the data.
/. (label)	Job step name on EXEC card such as //STEP1 EXEC .....
/& (end of job)	next // EXEC card or end of job or //
/* (end of data)	/* optional
/* (end of data)	/* optional
/+ (end of procedure)	// PEND
* (comment)	//* (comment)

For further details, see

- *CICS for VSE/ESA 2.3 System Control Statement, SC33-6613*.
- *OS/390 MVS System Commands, GC28-1781*.
- *OS/390 MVS JCL Reference, GC28-1757*.

for more information.

8.8.8 Rewrite CICS Macro-Level Programs

Migrate application programs from the CICS macro-level to the CICS command-level API. This section could be a book in itself, and there is not room here to explain all the programming techniques required. However, we present a summary of the important points to help you plan your reprogramming effort.

8.8.8.1 CICS Migration Aid

IBM supplies a program, ″The CICS Migration Aid,″ which you can use to convert the source of many user programs into the latest compiler version. However, notice that this is a fee product and is not free. See Appendix F, “Conversion Aid” on page 281 for more detail.

8.8.8.2 Identifying Macro-Level Programs

There is a standard CICS utility, DFHMSCAN, that scans user program sources and lists those programs that contain CICS macro calls and thus need to be rewritten for the command-level API. Run this utility against all source program libraries by running JCL or the like against program load libraries on VSE, as shown in Figure 65.

```
* $$ JOB
// JOB DFHMSCAN
// ASSGN SYS001,SYSLST
// ASSGN SYS002,SYSLST
// LIBDEF PHASE,SEARCH=(USER.PROGLIB,USER.PROGLIB2,CICS.LOADLIB)
// EXEC DFHMSCAN,PARM=′USER.PROGLIB,$SUMMARY′
/&
* $$ EOJ
```

Figure 65. Sample Job to Run DFHMSCAN against COBOL Program Load Library

This sample job will print out a list of the load modules that contain CICS macro-level code, and what language they were written in. You can also scan particular modules for further information by coding

```
// EXEC DFHMSCAN,PARM=′USER.PROGLIB,PROG1,PROG2........′
```

at the EXEC card.

8.8.8.3 What Must Be Recoded

The following is a list of the coding constructs users must rewrite into standard command-level CICS API statements for application programs, and CICS API, SPI and Gate macro calls for exits:

- EXEC CICS ADDRESS CSA
- DFHxx programming macro calls
- DFHxxx TYPE=DSECT calls, to map out control block storage.
- Calls to undocumented facilities such as DFHSEC macro calls.

A fuller comparison is available in Appendix D, “CICS Command-Level Equivalents of MACRO Calls” on page 273. The source must be migrated to MVS, edited, compiled, and linked on OS/390.

8.8.8.4 Copy the Source for All Current Programs

Copying application program sources from the VSE to the OS/390 environment is no different from copying CICS table sources that are described in 8.6.2, “Copy All CICS Tables” on page 89. We repeat some important points here but readers should refer to the that subsection for details of methodology and other information. The important points are these:

1. Allocate source library using ISPF with record size of 80.

2. Use jobs similar to the ones in Figure 66 on page 126 and Figure 67 on page 126 to create a tape containing the sources from the source library.

```
* $$  JOB JNM=SRCDUMP,CLASS=B,...
// JOB SRCDUMP accounting-info
// PAUSE   Pls. mount the tape
// TLBL SYSPCH,'VSE.USER.PGMLIB',,'TAP005'
// ASSGN SYSPCH,320
// LIBDEF PHASE,SEARCH=(PRD1.BASE,PRD2.BASE)
// EXEC LIBR
  ACCESS SUBLIB=USER.PROGLIB
  PUNCH PROGnnnn.C FORMAT=IEBUPDTE EOF=NO
  PUNCH .........C FORMAT=IEBUPDTE EOF=NO
  PUNCH .........C FORMAT=IEBUPDTE EOF=YES
/*
// MTC RUN,320
// ASSGN SYSPCH,UA
/&
* $$ EOJ
```

Figure 66. Sample Job to Transfer Application Programs Source to Tape

3. This tape is then copied to the PDS program source library in MVS.

```
//PGMLOAD  JOB PGMLOAD,'PGMRNAME',
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
// USER=IMSVSE2,NOTIFY=IMSVSE2,
// REGION=32M
//*
//STEP1    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   DSNAME=MVS.USER.PGMLIB,DISP=OLD
//SYSIN    DD   DSNAME=VSE.USER.PGMLIB,DISP=OLD,VOL=SER=TAP005,
//      LABEL=(2,BLP),DCB=BLKSIZE=80,UNIT=3490
/*
```

Figure 67. Sample Job to Unload Program Source from Tape to PDS

4. Amend the programs using the ISPF editor, then compile and link.

8.8.8.5 Translate, Compile, and Link Programs

JCL in CICS.SJCICSD.DBDC.JCL(COBCICS) shows the procedure by which DFHEITVL is delivered as standard with CICS into SYS1.PROCLIB. It is shown in Figure 139 on page 260 in Appendix B, "Resources Needed by Initial Installer" on page 239. Use this procedure for all application program work. Figure 68 on page 127 shows a sample job.

```
//STEP01 EXEC PROC=DFHEITVL,
//       INDEX=CICSTS12.CICS',
//       PROGLIB=<your.application.PROGLIB>
//TRN.SYSIN DD DISP=SHR,DSN=<your.application.srclib>
```

Figure 68. Sample of JCL Used to Run DFHEITVL Procedure

8.8.9 Rewrite Exits

Rewrite all exits to replace control block DSECTS with CICS gateway macros and CICS SPI calls. The autoinstall exit can probably work without change. For most other exits, the function changes depending on the release of CICS you are migrating from. See *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683*, for more information. The following constructs must be recoded:

- EXEC CICS ADDRESS CSA
- Addressing the CSA through R13
- DFHTC programming macro calls
- DFHxxx TYPE=DSECT calls, to map out control block storage
- Calls to undocumented facilities such as DFHSEC macro calls
- VSE abend code handling
- Explicit SVC calls
- Calls to MVS system macros
- Calls to POWER macros

8.8.10 Migrate from Local Security to External Security

CICS TS security is different in the CICS VSE environment. The following describe migration work.

8.8.10.1 CICS DL/I Security on VSE

Security has changed substantially since earlier releases of CICS. Users have defined security through ICCF, or by coding a sign-on table. In the sign-on table, the user has a hard-coded password, which allows access to a list of transaction classes. Users may also have implemented RSL levels. All files are automatically available to everyone, unless protected by RSLs.

8.8.10.2 CICS and IMS DBCTL Security on OS/390 with RACF

IBM supplies RACF for use on OS/390. It is normally installed as an integral part of the operating system.

In RACF you define Users, which are collected into groups. Everything else (transaction, file, connection, PSB, program, application, and so on) is resources that have RACF profiles defined. Users and groups must have permissions defined in RACF to be able to use those resources. RACF can return three possible values:

Allowed.
Not allowed.
Don't know.

You decide when you code the CICS and IMS start-up parameters what resources users will protect. A full set of parameters and definitions is given in Appendix E, “Security Parameters” on page 277

8.8.10.3 What Needs to be Defined

Users can issue all these commands from ISPF Option 6. Another way is to code them into a command file and execute it with EXEC ′filename′ from ISPF Option 6. The steps are these:

1. Check that the resource classes TCICSTRN, GCICSTRN, AIMS and LOGSTRM are defined to RACF.

   Issue the command:

   ```
   SETROPTS LIST.
   ```

   The output contains a list of all the resource classes in the system. If you don′t find the four resource classes, then get them defined in the RACF class descriptor table. This requires an IPL.

2. The groups IMS and CICS will have been set up as part of the original install. List them with:

   ```
   LG IMS
   LG CICS.
   ```

   If they are not found, follow the procedure in B.2.3, “RACF Definitions” on page 242 to define the groups.

3. Set up high level qualifiers for all database files. Define the catalog alias. Assume that ′ICBCHZ′ will be the HLQ for all future application databases, and it is cataloguing its data sets in CAT1.USERCAT. Thus, it should be:

   ```
   DEF ALIAS(NAME(′USER1) REL(′CAT1.USERCAT′)
   AG ICBCHZ OWNER(SYS1) SUPGROUP(SYS1)
   AD ′ICBCHZ.**′        GENERIC  OWNER(CICS) UACC(READ)
   AD ′ICBCHZ.??????.**′ GENERIC  OWNER(CICS) UACC(READ)
   PE ′ICBCHZ.??????.**′ ID(IMS)  ACCESS(UPDATE)
   PE ′ICBCHZ.**′        ID(IMS)  ACCESS(UPDATE)
   ```

4. Define the users under which the CICS and IMS-started tasks will run. This means you need to define the users for these tasks:

   - CICSD
   - IMSDDD
   - DBRC
   - DLISAS
   - IMSDRDR

   Two users should be enough. Enter this:

   ```
   AU IMSSTC  OWNER(IMS)  DFLTGRP(IMS)  NAME(′IMS  STARTED TASK USER′)
   AU CICDSTC OWNER(CICS) DFLTGRP(CICS) NAME(′CICS STARTED TASK USER′)
   ```

5. Define the STARTED resources to run the started tasks under these users:

```
RDEF STARTED IMSDDD.**  OWNER(IMS)  STDATA(USER(IMSSTC) GROUP(IMS))
RDEF STARTED DLISAS.**  OWNER(IMS)  STDATA(USER(IMSSTC) GROUP(IMS))
RDEF STARTED DBRC.**    OWNER(IMS)  STDATA(USER(IMSSTC) GROUP(IMS))
RDEF STARTED IMSDRDR.** OWNER(IMS)  STDATA(USER(IMSSTC) GROUP(IMS))
RDEF STARTED CICSD.**   OWNER(CICS) STDATA(USER(CICDSTC) GROUP(CICS))
```

6. Define all CICS users and groups to collect them in:

```
AG USERGP1 OWNER(ICBCHZ)  SUPGROUP(ICBCHZ)
AU FRED    OWNER(USERGP1) DFLTGRP(USERGP1) NAME('FRED CICS USER')  -
     CICS(OPCLASS(...) OPIDENT(FRD)) PASSWORD(INITIAL))
AU SALLY   OWNER(USERGP1) DFLTGRP(USERGP1) NAME('SALLY CICS USER') -
     CICS(OPCLASS(...) OPIDENT(SAL)) PASSWORD(INITIAL))
AU SING    OWNER(USERGP1) DFLTGRP(USERGP1) NAME('SING CICS USER')  -
     CICS(OPCLASS(...) OPIDENT(SIN)) PASSWORD(INITIAL))
```

7. Define all CICS transactions and give users permission to use them:

```
RDEF TCICSTRN CICD.CEMT OWNER(CICS) UACC(READ)
RDEF TCICSTRN CICD.JS** OWNER(CICS) UACC(NONE)
RDEF TCICSTRN CICD.C*** OWNER(CICS) UACC(NONE)
RDEF TCICSTRN CICD.CDBC OWNER(CICS) UACC(NONE)
RDEF GCICSTRN CICD.DFHCLS67 OWNER(CICS) UACC(NONE)       -
    ADDMEM(SABC,FFGT,SABD,SAXX,UITY,DFGT)
PE CICD.JS**    CLASS(TCICSTRN) ID(USERGP1) ACCESS(READ)
PE CICD.CDBC    CLASS(TCICSTRN) ID(IMS)     ACCESS(READ)
PE CICD.DHCLS67 CLASS(GCICSTRN) ID(USERGP1) ACCESS(READ)
```

   It is possible that users can handle a whole transaction class as a single unit by using the GCICSTRN class. Or, users can define each transaction individually using TCICSTRN records. However, users cannot define a transaction both by a TCICSTRN and as part of a GCICSTRN record.

8. Users may need more than one AGN. BMP0,1 which was set up initially, has access to everything. However, you may well decide to be more selective in future. For example, if users have several CICS systems whose databases they wish to keep apart, then you will want to define one per CICS system (and use its name in the DRA).

   The command to define a new AGN is

```
    RDEF AIMS BMPnn  OWNER(IMS) UACC(NONE)
    PE   AIMS BMPnn  ID(CICS)   ACCESS(READ)
```

9. Set the CICS start-up overrides SEC=YES and XTRAN=YES.
10. Set the IMS start-up parameter ISIS=1.
11. If users want more time to define all the transactions, then define all the users and set SEC=YES. This will at least force each user to sign on when logging on to the system. Define the transactions afterward, and set XTRAN=YES later.

8.8.11 Rewrite Programs to Make Them Year 2000 Compliant

This is an optional migration step and is covered in Appendix I, "Year 2000 Considerations" on page 293.

8.9 Postmigration Tasks

Some of these are optional and will not affect the main migration work. However, they should be done as soon as possible.

8.9.1 Install Full External Security

Install full external security using a security-server such as RACF. RACF is the product explicitly described in this book, but other products could be used to achieve the same results. Appendix E, "Security Parameters" on page 277 lists the resources the user should protect and the IMS and CICS parameters that the user can use, such as the Security Labels and Controlled Programs (nothing to do with APF list) features of RACF to add extra levels of security.

If an installation had RSLs implemented before, then you need to implement Security Labels to get a similar level of security.

8.9.2 Recode Batch DL/I Jobs as IMS BMPs

Batch jobs accessing the IMS database can continue to run as batch jobs; however, this is an inefficient way to work, and will continue to lock up resources in CICS. Such locking up can cause CICS to hang up when the batch job is either busy or is itself stopped waiting for resources. BMPs avoid much of this contention, as they share only the database records in question.

You need to recode EXEC DL/I statements (or DFHFC database calls) as GU/GN/ISRT/REPL (and other) calls coded using the CALL ASMTDLI, CALL CBLTDLI, or CALL PLITDLI interface depending on whether the applications are written in Assembler, COBOL, or PL1. Alternatively, just use

```
CALL CEETDLI
```

if the OS/390 LE interface has been installed.

Also, add a 32-byte **IOPCB** in the application programs, if it is not already coded. We extract parts of Figure 63 on page 119 and show them in Figure 69 on page 131 to illustrate the difference.

```
     ..............
 LINKAGE SECTION.

 01  IOPCBA POINTER.      1  IOPCB pointer
 01  DBPCB1 POINTER.      1  PCB pointer

**********************************************
* I/O PCB
**********************************************

 01  LTERMPCB           PIC  X(32).

**********************************************
* DATA BASE PCB
**********************************************

 01  DBPCB              PIC  X(50).

*****************************************************************
 PROCEDURE DIVISION USING IOPCBA, DBPCB1.  2  USING parameter
*****************************************************************
*
 SET ADDRESS OF LTERMPCB TO ADDRESS OF IOPCBA.   3  set pointer
 SET ADDRESS OF DBPCB TO ADDRESS OF DBPCB1.      3  set pointer
```

Figure 69. Sample BMP Program

The BMP program should also issue checkpoint (CHKP function call) from time to time to release resources.

You will also need to change JCL to use the IMSBATCH procedure to run the program; see Figure 70.

```
//IMSBMPXX JOB (accounting info),'IMS BMP',CLASS=a,MASCLASS=T
//*
// JCLLIB ORDER=IMS.SJIMSD.PROCLIB
//STEP01  EXEC IMSBATCH,
//          MEMBER=program-name,
//          PSB=psb-name,
//          IMSID=IMSD,
//          AGN=BMP01,
//          APARM='program parameter data'
//STEPLIB   DD
//          DD
//          DD DISP=SHR,DSN=your-program-library
//
```

Figure 70. Sample IMSBATCH Procedure to Run BMP

Before you can run the BMP, you must also have to define the program to IMS (DBCTL online region in our case), by coding an entry in the application copybook for the PSB in member IMS.SJIMSD.STAGE1.SOURCE(APPLDB), such as:

```
APPLCTN PSB=yourbmp,PGMTYPE=BATCH
```

and running an IMS MODBLKS generation. See 8.7.6, "Run IMS MODBLKS Generation" on page 109.

8.9.3 Improve Sequence Processing

If application programs do a lot of sequential retrieval, you may want to consider using DB Tools HSSR call interface instead of the standard DL/I call. The following is an example:

```
Replacing all:

    ...........
    CALL 'CEETDLI' USING .....
by:

    ...........
    CALL 'CBLHSSR' USING .....
```

Users been reported that it can reduce sequential retrieval time by up to 4 times. Reference *IMS System Utilities/Data Base Tools, High Speed Sequential Retrieval Guide, SH21-0548* for more information. HSSR is a fee package, not free.

8.10 Optional Migration Tasks

IBM strongly advises that you perform the steps described in this section. However, they are not required at the release levels described in this book, and can be left to a later date if you prefer.

8.10.1 Migrate CICS DCT Table to RDO

This is very simply done and could well be done at the same time you migrate the FCT. However, the DCT migration is put here because it is optional at CICS/TS 1.2.

When you elect to do it, change the DCT source so that it can be assembled and migrated to RDO, as follow:

- Using the ISPF editor, change the DCT source in CICS.SJCICSD.TABSRC(DFHDCTD0):
  - On the

    ```
    DFHDCT TYPE=INITIAL
    ```

    macro,

    change the suffix to

    ```
    SUFFIX=MG
    ```

  - Change the subparameter

    ```
    TYPE=INITIAL
    ```

    to

    ```
    TYPE=(INITIAL,MIGRATE)
    ```

- Add an entry:

  ```
  DFHDCT TYPE=GROUP,GROUP=APPLDCT
  ```

Then assemble and migrate using a job such as that shown in Figure 71.

```
//DCTRDO  JOB (accounting-info),'Migrate DCT to RDO',
//        CLASS=A,MSGCLASS=3
//*
//DCTASM  EXEC TABASM
//C.SYSIN   DD DISP=SHR,DSN=CICS.SJCICSD.PARMLIB(DCTDO)
//L.SYSLMOD DD DISP=SHR,DSN=CICS.SJCICSD.TABLOAD
//L.SYSLIN  DD
//          DD *
        NAME DFHDCTMG(R)
/*
//FCTMIGR EXEC PGM=DFHCSDUP
//STEPLIB   DD DISP=SHR,DSN=CICS.SJICSD.SDFHLOAD
//          DD DISP=SHR,DSN=CICS.SJICSD.TABLOAD
//DFHCSD    DD DISP=SHR,DSN=CICS.SJCICSD.DFHCSD
//SYSIN     DD *
  MIGRATE TABLE(DFHDCTMG)
  COPY GROUP(APPLDCT) TO(APPLGRP)
  VERIFY
  LIST ALL
/*
//
```

Figure 71. Sample JCL to Load DCT Definitions into CSD

Once this has been done, install the group APPLDCT and then delete it (because the contents have been copied into a group already on the start-up list). Use the commands on-line in CICS:

```
CEDA INSTALL GROUP(APPLDCT)
CEDA DELETE  ALL(*) GROUP(APPLDCT)
```

As an alternative, a cold start of CICS will also install the resources.

Finally change the SIT override member

`SJCICSD.PARMLIB(OVERRIDES)`

so that the line

`DCT=DO`

becomes

`DCT=NO`

.

8.10.2 Convert Terminals and Programs to Autoinstall

You can do this a bit at a time. Once autoinstall is activated, CICS continues to use a definition if it finds one, and creates one only if one does not already exist.

8.10.2.1 Autoinstall for Terminals

You may already have an autoinstall exit for terminals, if so you should be able to compile it on OS/390 and use it. Autoinstall model terminals will also have been copied over to OS/390. It seems everything should work. In practice, however, it does not, and you may have to spend time fiddling with model terminal definitions to make it work in all cases, as the matching criteria become stricter with each release, and even at different maintenance levels of the same release. Do not despair: this is the sort of problem you are employed to fix. Combining a knowledge of all the bits in a LOGMODE table with trial and error can get this working.

If you do not currently use autoinstall, you should consider moving to it. It can save an enormous amount of storage at run time, as it loads control blocks for only the terminals in use. It can also save substantially on restart time. If you have no conventions for CICS terminal name, and all terminals are of the same type, you can use the default autoinstall program. The default action of the sample program, at INSTALL, is to select the first model in the list, and derive the terminal identifier from the last four nonblank characters of the LU name, set the status byte, and return to CICS. If there are no models in the list, it returns with no action.

If you have a particular naming convention for CICS terminals, you can either code logic in the autoinstall program to generate the names according to convention, or else read the names from a file. If you have particular model terminal definitions that users want included, or if they wish to scan a whole list, then include this logic in your autoinstall program.

You can start with the samples provided by IBM and modify them. The samples provided are in CICSTS.CICS.SDFHSAMP, and are summarized in Table 13.

Table 13. IBM Supply Autoinstall Terminal Programs

Language	**Sample source name**	**Location of copybook**
Assembler	DFHZATDX	SDFHMAC
COBOL	DFHZCTDX	SDFHCOB
PL/I	DFHZPTDX	SDFHPLI
C/370	DFHZDTDX	SDFHC370

For further details refer to *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683*. Set the name of the autoinstall program using the CICS start-up parameter AIEXIT=.

8.10.2.2 Autoinstall for Programs

The introduction of program autoinstall ends the need to define to CICS the programs to be used. As with autoinstall of terminals, users supply model definitions in the CSD and the autoinstall program attempts to use one of these to create a definition it can use to load the application program. However, unlike terminal autoinstall, autoinstall for programs is unlikely to need a large number of models. Indeed, for many sites one model would probably suffice; you are likely to find that the IBM-supplied module DFHPGADX is adequate.

If users want to write their own programs, the names of the IBM samples are shown in Table 14 on page 135.

Table 14. IBM Supply Autoinstall Programs		
Language	**Name of member**	**Library**
Program source:		
Assembler	DFHPGADX	SDFHSAMP
COBOL	DFHPGAOX	SDFHSAMP
PL/I	DFHPGALX	SDFHSAMP
C/370	DFHPGAHX	SDFHSAMP
Copy books:		
Assembler	DFHPGACD	SDFHMAC
COBOL	DFHPGACO	SDFHCOB
PL/I	DFHPGACL	SDFHPL1
C/370	DFHPGACH	SDFHC370

For further details, see the manual *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683*.

8.10.3 Convert to 31-Bit Addressing

Traditionally, most of the application programs in the VSE environment use 24-bit addressing. The 24-bit address load modules can do their processing only in the user area below the 16 MB boundary of an MVS address-space. Compared with a total of 2 GB MVS address space, 16 MB is a small area (refer to Figure 6 on page 12). We recommend that you convert applications to 31-bit addressing whenever possible, especially on-line applications.

Converting to 31-bit addressing is relatively easy. However, you need to convert the whole unit at one time. This means that for a load module statically linked with the main program and many subroutines, the main program and all subroutines must all be converted to 31-bit addressing. Mixing of 24- and 31- bit addressing is not allowed. These are the methods to convert to 31-bit addressing:

- **Assembler programs**. Insert two statements, or change them if they already exist, in the source program as follows:

```
         AMODE  31
         RMODE  ANY
```

  Address mode,

  `AMODE 31`

  indicates 31-bit addressing. Resident mode,

  `RMODE ANY`

  means that the load module could reside anywhere in the MVS address space, either below 16 MB or above it in the user area. The application then looks like:

```
APPLNAME CSECT
         AMODE  31
         RMODE  ANY
BEGIN    SAVE   .......
         USING  BEGIN,15
        ...........
```

```
     <application code>
        ...........
        L      13,SAVEAREA+4
        RETURN (14,12)
SAVEAREA DS     18F
OPTIONS  DC     Y(OPTIONSL)
OPTS     DC     C'XREF(SHORT)'
     <more compiler options>
OPTIONSL EQU    *-OPTS
        END
```

- **For COBOL programs**, use compiler option DATA(31).
- **At linkage editor step**. Add parameters AMODE and RMODE on the execution statement as shown:.

```
//LKED   EXEC PGM=IEWL,REGION=.....,
//       PARM='AMODE=31,RMODE=ANY,<other parameters>'
```

- **Run-time (batch programs)**. Data sections can be allocated above 16 MB by using run-time parameter ALL31(ON), following this example:

```
//STEPnn  EXEC  PGM=APPLPGM,PARM='ALL31(ON)/<others>'
//        <other JCL as needed>
//
```

8.10.4 Using a Data Table

Many application systems, SAFE II included, assemble the most frequently used data, or short subroutines, into an assembler DSECT and compile it as a load module. The module is loaded into system user area. When required, the application finds the offset location of the module, loads the data or branch to that location, executes the code, and returns. This allows data to be accessed at memory speed instead of I/O speed by other means. In a short and simple subroutine, this saves program management time overhead.

For larger sets of data that are referred to frequently, a VSAM keyed sequential file is probably most used to access information.

These type of frequently and mostly read-only accessed databases are the best candidates to convert to the CICS DATA TABLE. CICS loads these tables into a data space at initialize time, and data then can be accessed at memory speed with very little I/O. The data tables can also be shared with many CICS regions. Refer to *CICS Transaction Server for OS/390 CICS Shared Data Table Guide, SC33-1702* for more information.

8.11 SAFE II Migration

Time constraints prevented us from migrating the entire SAFE II application system to OS/390 from VSE during the residency project. Instead, we selected the most frequently used programs, both on-line transactions and batch programs. We converted these into the OS/390 environment with the methods already discussed, installed them, and tested them on our ITSO OS/390 facility. The result proved successful. We were able to simulate most of the daily operation by the end of the project.

We documented a number of results for reference. These might be useful for similar migration projects in the future, since SAFE II does have a large installed base, not only in China but in many parts of the world. These findings are application specific and may or may not apply to other systems.

8.11.1 Data-Only Load Module

SAFE II uses data-only load modules to store and access frequently used data and subroutines (see 8.10.4, “Using a Data Table” on page 136 for more information on the data-only module). We found that load-module prefix length, generated by the CICS command-level language translator, is different between CICS VSE and CICS TS. Because CICS TS generates a prefix 8 bytes longer than CICS VSE, all calling programs that calculate offset must add another eight positions to be able to get the right location.

8.11.2 Default PSB Name

Many of the SAFE II programs did not code the PSB name. At the *SCHEDULE PSB* time, the PSB name was picked up from the ACT table, which is not supported by CICS TS. Therefore, we have to add the PSB name into all programs so that it schedules the right PSBs during processing.

8.11.3 IOPCB Structure

We added the IOPCB data structure into all selected batch programs so that we can run BMP as well as stand-alone batch jobs. The testing results are fully satisfactory and provide greater flexibility for future production scheduling.

We also added parameters CMPAT=YES in PSB, which will allow the same PSB to run both as stand-alone batch as well as BMP.

8.11.4 Access to JCT

After careful study of the SAFE II codes, we found that access to JCT to verify the existence of journal files is not necessary for the OS/390 MVS log stream. We therefore deleted the code. However, redesign to use new CICS TS API is under way.

Chapter 9. Operating IMS DBCTL

Operating CICS with DBCTL presents opportunities that are not available when running DL/I locally in a CICS region. For example, CICS with DBCTL can start and stop database and on-line regions independently. Also you can connect a CICS region to a different set of databases (for example to switch between two training environments, so you can use one while maintaining the other) simply by issuing a couple of CICS commands.

We do not cover the issue of operation automation here, as it depends very much on what automatic operation (AO) product is installed and what automation policies are decided on. That would be a redbook in itself.

9.1 Running CICS TS

We describe CICS TS operation in this section. CICS TS operation is mostly the same as in CICS VSE.

9.1.1 Starting CICS TS

Normally, CICS TS can be started from the OS/390 system console by issuing the command:

```
START CICSD
```

There is a START= parameter which can be INITIAL, COLD, WARM, EMERGENCY, or AUTO. The initial default is INITIAL, but change this default to AUTO if you install CICS logging. These options are described later, in 9.6, “Restart” on page 157. CICS is available for logon when the message

```
+DFHSI1517 SCSCPAAD Control is being given to CICS.
```

appears on the console, where SCSCPAAD is the VTAM APPLID of the CICS region.

Part of the initialization process is to initialize the CICS log stream with messages (displayed over the console) that connect to the OS/390 log stream as follows:

```
+DFHLG0104I SCSCPAAD 224
 System log (DFHLOG) initialization has ended. Log stream
 CICS.SJCICSD.DFHLOG is connected to structure ****************.
+DFHLG0103I SCSCPAAD System log (DFHSHUNT) initialization has started.
+DFHLG0104I SCSCPAAD 226
 System log (DFHSHUNT) initialization has ended. Log stream
 CICS.SJCICSD.DFHSHUNT is connected to structure ****************.
+DFHLG0102I SCSCPAAD Log manager domain initialization has ended.
```

As in CICS/VSE, you can code a list of programs in a start-up PLT and have them called during Stages 2 and 3 of CICS start-up. These allow you to execute programs to initialize other products and applications. The link to IMS DBCTL can be started up in this way. Since you no longer use a sequential terminal to issue CSMT commands, you may wish to initialize resources this way using a user-written PLT program.

9.1.2 Stopping CICS TS

To shut down CICS TS while still allowing all outstanding transactions to complete, issue the command:

```
    CEMT PERFORM SHUTDOWN
or  CEMT P SHUT
```

This can be entered from any console or terminal with the authority necessary to issue the command.

To shut down CICS/TS quickly, abending any unfinished transactions, issue the command:

```
    CEMT PERFORM SHUTDOWN IMMEDIATE
or  CEMT P SHUT IMM
```

Again, this can be entered from any console or terminal given the authority necessary to issue the command.

There are also other forms of the shutdown command, unavailable with earlier versions of CICS, which users can use in special circumstances. For example,

```
    CEMT PERFORM SHUTDOWN SDTRAN(ssss) PLT(xx) XLT(xx)
```

The three optional parameters, SDTRAN, PLT, and XLT allow you to specify a PLT, XLT or SDTRAN which then replaces those specified in the SIT overrides. Alternatively, you can specify NOSDTRAN, which closes without using any shutdown transaction. These three options are:

- **Shut down PLT**. Now, with two phases separated by the entry DFHDELIM, shut down PLT is the same as in VSE. It allows CICS to run programs to disconnect other resources and applications from CICS in a clean manner. SIT override

  PLTSD=xx

  defines the suffix of the PLT to be used to shut down CICS.

- **XLT**. This is a list of transactions that you issue during Stage 1 shutdown. CEMT and CESF are always allowed, as are all transactions defined with SHUTDOWN(ENABLED), regardless of what is defined in the XLT. You specify the suffix of the XLT to use with the SIT override

  XLT=

- **SDTRAN**. This allows you to write a transaction to execute during shutdown. It can be used to purge long-running transactions. You specify this transaction with SIT override SDTRAN=.

Since users can now specify special shutdown parameters, there are two ways this facility can be used:

- Test new versions of PLT, SDTRAN, or XLT before coding them in SIT. This is easy to back out since the backout is simply to issue the command without the extra parameters.

- Keep special versions for special circumstances.

9.1.3 Stopping CICS Transactions

If users find that transactions do not close during shutdown, users can identify them and then purge them thus:

```
CEMT I TA     <=== To get a list of tasks.
```

These will be listed like this:

```
I TA
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tas(0000858) Tra(CEMT) Fac(NM18) Run Ter Pri( 255 )
   Sta(TO) Use(IMSVSE1 ) Uow(B060DC584EFF0507)
Tas(0000877) Tra(XPDF) Fac(NM07) Run Ter Pri( 255 )
   Sta(TO) Use(IMSVSE5 ) Uow(A060DC585EFF6504)
```

The numbers in paraentheses by TAS(nnnn) are task IDs. These can be purged by issuing the command:

```
CEMT S TA(877) PURGE
```

where the TA(877) is the task ID found from the CEMT I TA. If this task is not purged, then a stronger form can also be used:

```
CEMT S TA(877) FORCE
```

9.2 Running IMS/ESA DBCTL

Operating IMS DBCTL online region may be new to most of the VSE (or earlier CICS versions) users. We are going to give a bit of detail in the following sections. Users should be aware that although there is an IMS master console in existence for the full IMS DM/TM online system, IMS DBCTL is not able to use it. Most of the IMS commands have to enter via OS/390 master console, SDSF or CICS supply transaction CDBM.

9.2.1 Starting IMS/ESA DBCTL

IMS DBCTL can also start from the OS/390 system console by issuing the command:

```
START IMSDDD
```

You can also specify AUTO=N or AUTO=Y. The initial default is N, but once IMS is running normally, you may prefer to set it to Y. If the start is set to AUTO=N, IMS performs its initial start until the message

```
*DFS989I IMS (DBCTL) READY (CRC=>) - IMSD
```

is displayed on the console, where IMSD is the *IMSID*, and > in CRC=> is a character that can be used instead of *IMSID* to prefix IMS commands issued from the console (see 9.4, "Operating IMS/ESA DBCTL" on page 147) below. Users then enter the command:

```
IMSDNRE.    <=== Always include the full stop.
```

from the system console for a warm start of IMS. IMS is available for use when this message appears:

```
DFS994I WARM START COMPLETED.
```

If users are performing a cold start, the restart command is instead

```
IMSDNRE CHECKPOINT 0.
```

and the response message issued is

```
DFS994I COLD START COMPLETED.
```

In all these cases the IMSD represents the *IMSID* of the system, which in our example was IMSD.

9.2.2 Starting CICS TS and IMS/ESA DBCTL Together

If you are starting CICS TS and IMS/ESA DBCTL at the same time, do it in the following order:

1. Activate ACT VTAM major node for CICS/TS (if not already activated).
2. Ensure that parameter DBCTLCON=YES is included in the start-up parameters file.
3. Start IMS/ESA DBCTL.
4. Start CICS TS.

9.2.3 Stopping IMS/ESA DBCTL

IMS should be closed from a system console. To close IMS in an orderly fashion, type

```
IMSDCHE PURGE.
```

To terminate IMS threads immediately, type

```
IMSDCHE FREEZE.
```

If threads or BMPs do not close, you can identify them and then purge them. First, display all regions and CICS threads by typing

```
IMSDDIS A.
```

This gives a display on SYSLOG and in the job log that resembles the following example:

```
DFS000I MESSAGE(S) FROM ID=IMSD
    REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS
        7 FREDBMP   BMP             FREDPROG
          FPRGN     FP    NONE
        3 CICSD     DBT   CICSD              AVAILABLE
        2 CICSD     DBT   CICSD              AVAILABLE
        1 CICSD     DBT   CICSD              AVAILABLE
          DBRC      DBRC
          DLISAS    DLS
    *98119/191940*
```

Look at the REGID in the first column. You must specify the REGID to purge the region; for example, to stop FREDBMP, type

```
IMSDSTOP REGION 7.
```

Or, to purge a CICS thread, type

```
IMSDSTOP REGION 3.
```

For a stronger form of purge — to be used only if the first form does not work — type

```
IMSDSTOP REGION 3 ABDUMP.
```

This will produce an MVS formatted dump of Region 3 (the CICS thread).

9.3 Operating CICS TS

Commands for CICS TS can be entered by the following means:

- From the system console, using the CEMT or the CEST transaction.
- From a terminal, using the CEMT or the CEST transaction.
- From a terminal, using CECI EXEC CICS INQUIRE/SET commands.
- From a user program that issues EXEC CICS INQUIRE/SET commands.

Note that the CSMT transaction to issue commands from a sequential terminal is not available in CICS TS.

CICS-supplied transactions should be very familiar to users, and so we give only a brief description here.

9.3.1 CEMT

CICS TS allows you to enter commands using the CEMT transaction, either from an SNA terminal or through the system console. You must sign on for either method. Also, you can use a remote terminal and route to the CICS TS system by using CRTE. Using CEMT, you can control all the resources of CICS, including deletion of resources (until the next cold start). Thus, you can connect and disconnect terminals, refresh programs, purge tasks, stop and start transactions, and set MAXTASKS. To put a terminal out of service from both a terminal and the console, use the sequence described here.

First, you must sign on by issuing the CESN command. When the sign-on screen appears, fill in the user ID and password. To sign on from a console, type the command

```
/F CICSD,CESN USERID=uuuuuuu,PS=pppppppp
```

where CICSID is the name of the task or job CICS is running as. Then clear the screen and issue this command

```
CEMT I TERM(NM07)
```

The resource name (such as a terminal) can contain generic characters. Then, Figure 72 on page 144 appears.

```
 I TERM(NM07)
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Ter(NM07)           Pri( 000 ) Pag  Ins  Ati Tti Loc
     Net(SCGPVM07) Rel



















                                                        SYSID=CICD APPLID=
  RESPONSE: NORMAL                                  TIME:  21.25.25  DATE: 0
PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 72. Terminal Display Responding to CEMT Command

Overtype the

INS

with

OUT

to set the terminal out of service. The result is the display shown in Figure 73.

```
 I TERM(NM07)
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Ter(NM07)           Pri( 000 ) Pag  Out  Ati Tti Loc          NORMA
     Net(SCGPVM07) Rel



















                                                        SYSID=CICD APPLID=S
  RESPONSE: NORMAL                                  TIME:  21.28.02  DATE: 04
PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 73. Set a Terminal Out of Service

Press PF3 followed by CLEAR to exit this screen.

You can also issue the commands from the terminal in their full form instead of overtyping the screens. Type this:

```
CEMT I TERM(NM07)
```

followed by

```
CEMT S TERM(NM07) OUT
```

The screen displays you see are the same as Figure 72 on page 144 and Figure 73 on page 144. If you are working at the system console, type

```
/F CICSD,CEMT I TERM(NM07)
```

followed by

```
/F CICSD,CEMT SET TERM(NM07)  OUT
```

You then see the screens as shown in Figure 72 on page 144 and Figure 73 on page 144, but on SYSLOG and in the CICS joblog.

9.3.2 CEST

You can also use the transaction CEST to issue commands to control terminals, lines, netname, and tasks. This capability is available from 3270 devices or from the system console. If you use the console, issue the command with the same syntax as for the CEMT equivalent, but append the parameter SUPRID(D5) where D5 is the suffix of a TLT (an on-line region can have as many TLTs as needed). CEMT can also be used with this SUPRID parameter in the same way. Type

```
CEST I TERM(NM07) SUPRID(D5)
```

followed by

```
CEMT S TERM(NM07) OUT SUPRID(D5)
```

9.3.3 CECI

CICS TS allows you to enter any EXEC CICS command through the CECI transaction. However, you must do so from a terminal, not the system console. This includes commands to control named resources, as well as to define and delete them (until the next cold start of CICS). You can also ask for and change the characteristics of a terminal on-line. However, CECI cannot handle generic names. Type

```
CECI INQUIRE TERM(NM07)
```

This yields the screen shown in Figure 74 on page 146.

```
INQUIRE TERM(NM07)
STATUS:  ABOUT TO EXECUTE COMMAND                          NAME=
 EXEC CICS  INquire TErminal( 'NM07' )
  < NETname() >
  < STart | ENd | NEXT >
  < ACCessmethod() >
  < ACQstatus() >
  < ALTPAGEHt() >
  < ALTPAGEWd() >
  < ALTPRInter() >
  < ALTPRTcopyst() >
  < ALTSCRNHt() >
  < ALTSCRNWd() >
  < ALTSUffix() >
  < APLKybdst() >
  < APLTextst() >
  < AScii() >
  < ATistatus() >
  < AUDalarmst() >
+ < AUToconnect() >



PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 74. Inquiry as to Defined Attributes of a Terminal

Six further screens then appear with empty parameter fields. If you press Enter, the values are all filled in as shown in Figure 75.

```
INQUIRE TERM(NM07)
STATUS:  COMMAND EXECUTION COMPLETE                        NAME=
 EXEC CICS  INquire TErminal( 'NM07' )
  < NETname( 'SCGPVM07' ) >
  < STart | ENd | NEXT >
  < ACCessmethod( +0000000060 ) >
  < ACQstatus( +0000000070 ) >
  < ALTPAGEHt( +00000 ) >
  < ALTPAGEWd( +00000 ) >
  < ALTPRInter( '    ' ) >
  < ALTPRTcopyst( +0000000447 ) >
  < ALTSCRNHt( +00000 ) >
  < ALTSCRNWd( +00000 ) >
  < ALTSUffix( '.' ) >
  < APLKybdst( +0000000392 ) >
  < APLTextst( +0000000394 ) >
  < AScii( +0000000001 ) >
  < ATistatus( +0000000075 ) >
  < AUDalarmst( +0000000395 ) >
+ < AUToconnect( +0000000171 ) >


 RESPONSE: NORMAL                EIBRESP=+0000000000 EIBRESP2=+0000000
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 75. Reset Terminal Attributes

As with CEMT, you can then overtype those you want to change.

You can also change a field by issuing the commands in full. For example, to change the NETNAME, type

```
CECI INQUIRE TERM(NM07)
CECI SET     TERM(NM07) ('SCGPNM55')
```

9.3.4 User Programs

EXEC CICS INQUIRE and EXEC CICS SET can be called from user programs, which can either be defined to CICS as transactions or run as PLT programs.

It has long been common practice to write simple programs to invoke DFHEMT to issue operator commands more commonly issued by means of CEMT but with restrictions. This can now be done by using the EXEC CICS INQUIRE / SET verbs, as well as the DEFINE and DISCARD verbs to perform temporary definition and deletion of resources.

In addition, it is now possible to run such programs in the start-up PLT in Stage 3 and pass it a parameter string using the SIT overrides INITPARM=(PLTPROG1='PARAMETERS',PLTPROG2='PARAMETERS') where pltprog1 and pltprog2 are the names of PLT programs, that retrieve the parameters using the command EXEC CICS ASSIGN INITPARM.

9.3.5 CSMT

The CSMT command, which many sites used to use to issue commands to CICS immediately after initialization, is not supported by CICS TS. You must instead write a program to issue the command and include it in the start-up PLT in Stage 2. You can pass parameters to this program using the SIT override. However, you can do this to only one instance of the program; that is, calling the program twice with different parameters is not allowed.

9.4 Operating IMS/ESA DBCTL

Commands for IMS/ESA DBCTL can be entered by the following means:

- Prefix it with the IMSID from the system console.
- Prefix it with the CRC from the system console.
- Prefix it with '/' using the CDBM transaction at a CICS terminal.
- Issue DBRC commands by any of the above methods using the /RMx commands.
- Issue DBRC commands using a batch job.
- Issue commands through the ICMD interface from a batch program.

The following subsections describe these methods in detail.

9.4.1 Issue Commands Using IMSID from System Console

You can issue any valid IMS DBCTL command through the system console by this method. Valid commands are listed in Appendix G, "Available IMS Commands for DBCTL" on page 287. Users should always employ this method to issue commands to start and stop IMS DBCTL. The IMSID is defined in SYSGEN, but can be overridden in the IMS start-up parameters. IMSID is always displayed in messages, so it is not easy for an operator to be confused over which IMS the command is destined for. In this case, it is *IMSD*. If the ID is not unique, then IMS will not start.

For example, to issue a command to display all OLDS, type

```
    IMSDDIS OLDS.
  or from SDSF /IMSDDIS OLDS.
```

This produces the display shown in Figure 76 on the SYSLOG and DBCTL joblog.

```
DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=IMSVSE1: DIS OLDS IMSD
DFS000I MESSAGE(S) FROM ID=IMSD
    OLDS-DDNAME % FULL RATE ARCH-JOB    ARCH-STATUS  OTHER-STATUS
    *DFSOLP01       0     0                            IN USE
    *DFSOLS01       0     0                            IN USE
     DFSOLP00                ARCIMSD     SCHEDULED
     DFSOLS00                            SCHEDULED
     DFSOLP05                            AVAILABLE
     DFSOLS05                            AVAILABLE
     DFSOLP04                            AVAILABLE
     DFSOLS04                            AVAILABLE
     DFSOLP03                            AVAILABLE
     DFSOLS03                            AVAILABLE
     DFSOLP02                            AVAILABLE
     DFSOLS02                            AVAILABLE
     DUAL OLDS LOGGING, DUAL WADS LOGGING
     AUTOMATIC ARCHIVE = 01
      WADS = *DFSWADS0 *DFSWADS1 DFSWADS2
     *98119/224902*
```

Figure 76. Display When Using IMSID from the System Console

9.4.2 Issue Commands Using the CRC from System Console

You can issue any valid IMS DBCTL command through the system console by this method. Valid commands are listed in Appendix G, “Available IMS Commands for DBCTL” on page 287. The CRC (command recognition character) is a single character used to prefix all commands. It is then used by MVS to route the command to the right IMS region. It is specified in IMS generation and can be overridden in the IMS start-up parameters. Operators and support staff can find it difficult to remember which CRC is defined for which subsystem, especially when there are several IMS (and DB2) subsystems on a machine. The CRC routing will not work if the chosen CRC is already in use when IMS starts, or if is in use by MVS (for example, ′$′ is used for JES2). The default is ′>′, which is used in this example. In general, we advise using the IMSID instead. For example, to display all OLDS, type

```
    >DIS OLDS.
  or from SDSF />DIS OLDS.
```

This produces the display shown in Figure 77 on page 149, on the SYSLOG and DBCTL joblog.

```
DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=IMSVSE1: DIS OLDS IMSD
DFS000I MESSAGE(S) FROM ID=IMSD
    OLDS-DDNAME % FULL RATE ARCH-JOB    ARCH-STATUS  OTHER-STATUS
    *DFSOLP01       0     0                           IN USE
    *DFSOLS01       0     0                           IN USE
     DFSOLP00               ARCIMSD     SCHEDULED
     DFSOLS00                           SCHEDULED
     DFSOLP05                           AVAILABLE
     DFSOLS05                           AVAILABLE
     DFSOLP04                           AVAILABLE
     DFSOLS04                           AVAILABLE
     DFSOLP03                           AVAILABLE
     DFSOLS03                           AVAILABLE
     DFSOLP02                           AVAILABLE
     DFSOLS02                           AVAILABLE
     DUAL OLDS LOGGING, DUAL WADS LOGGING
     AUTOMATIC ARCHIVE = 01
      WADS = *DFSWADS0 *DFSWADS1 DFSWADS2
     *98119/224902*
```

Figure 77. DIS OLDS Display Using CRC from System Console

9.4.3 Issue Commands to IMS from CDBM at a CICS Terminal

This allows CICS users to issue most IMS commands at a screen that looks similar to an IMS/TM on-line screen. Some commands, such as /CHE and /MOD which close IMS or install new resource definitions, are not available through CDBM.

For example, to display OLDs using CDBM, first enter CDBM at any CICS terminal as shown in Figure 78 on page 150.

```
 CDBM                  CICS-DBCTL Operator Transaction


 Type IMS command.






 For /DBDUMP or /DBRECOVER commands

 Choose one. 1 1. Do not force end of volume
               2. Force end of volume


 Press enter to display responses.


 CICS APPLID SCSCPAAD
 DBCTL ID    IMSD


F1=Help  F3=Exit  F5=Refresh  F12=Cancel
```

Figure 78. Screen after Entering the CDBM Transaction

Type in the command you want (/DIS OLDS. as shown in Figure 79) and press Enter.

```
 CDBM                  CICS-DBCTL Operator Transaction


 Type IMS command.
             /DIS OLDS.





 For /DBDUMP or /DBRECOVER commands

 Choose one. 1 1. Do not force end of volume
               2. Force end of volume


 Press enter to display responses.


 CICS APPLID SCSCPAAD
 DBCTL ID    IMSD


F1=Help  F3=Exit  F5=Refresh  F12=Cancel
```

Figure 79. CDBM Transaction Input Screen

The result is displayed in Figure 80 on page 151.

```
 CDBM                     CICS-DBCTL IMS Responses                Screen 1
                                                          Responses 1    to 17
                                                                      More
S OLDS

 *DFSOLP01       0     0                          IN USE
 *DFSOLS01       0     0                          IN USE
  DFSOLP00               ARCIMSD     SCHEDULED
  DFSOLS00                           SCHEDULED
  DFSOLP05                           AVAILABLE
  DFSOLS05                           AVAILABLE
  DFSOLP04                           AVAILABLE
  DFSOLS04                           AVAILABLE
  DFSOLP03                           AVAILABLE
  DFSOLS03                           AVAILABLE
  DFSOLP02                           AVAILABLE
  DFSOLS02                           AVAILABLE
  DUAL OLDS LOGGING, DUAL WADS LOGGING
  AUTOMATIC ARCHIVE = 01
  WADS = *DFSWADS0 *DFSWADS1 DFSWADS2
 *98119/230821*

DFHDB8228 The period (.) and subsequent characters have been removed.
F1=Help  F3=Exit  F4=Top  F6=Bottom  F7=Bkwd  F8=Fwd  F9=Retrieve  F12=Cancel
```

Figure 80. CDBM Response Screen

See *CICS Transaction Server for OS/390 CICS IMS Database Control Guide, SC33-1700*, for more information.

9.4.4 Issue DBRC Commands to IMS Online

In this example, the commands are issued from the system console with IMSID. We give the appropriate syntax for the other methods also. The online form is to use a command RMx DBRC='yyyyyy...' where the x is the first letter of the DBRC command, and the part inside the single quotes is the whole command after the '.' for example, LIST.RECON STATUS would become RML DBRC='RECON STATUS'. This is issued using IMSID from a console by typing

```
   IMSDRML DBRC='RECON STATUS'.
or from SDSF as /IMSDRML DBRC='RECON STATUS'.
```

This is issued using CRC from a console by typing

```
   >RML DBRC='RECON STATUS'.
or from SDSF as />RML DBRC='RECON STATUS'.
```

This is issued using CDBM from a CICS terminal by typing

```
   /RML DBRC='RECON STATUS'.
```

The result is displayed as shown in Figure 81 on page 152.

```
 DFS000I MESSAGE(S) FROM ID=IMSD
 LIST.RECON STATUS
 -------------------------------------------------------------------- +
 ------
  RECON
   RECOVERY CONTROL DATA SET, IMS/ESA V6R1
   DMB#=0                             INIT TOKEN=98111F2015518F
   NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
   TAPE UNIT=3490      DASD UNIT=3390      TRACEOFF   SSID=IMSD
   LIST DLOG=NO                 CA/IC/LOG DATA SETS CATALOGED=YES
   LOG RETENTION PERIOD=00.001 00:00:00.0
   TIME STAMP INFORMATION:
    TIMEZIN = %SYS
    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY
 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMS.SJIMSD.RECON1
  RECON2        COPY2          IMS.SJIMSD.RECON2
  RECON3        SPARE          IMS.SJIMSD.RECON3
DSP0180I  NUMBER OF RECORDS LISTED IS        1
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 98.119 23:39:52.1
DSP0211I  COMMAND PROCESSING COMPLETE
DSP0211I  HIGHEST CONDITION CODE = 00
DSP0058I  RML COMMAND COMPLETED
```

Figure 81. Response of /RML DBRC=′RECON STATUS′ Command

See *IMS/ESA, Operations Guide, SC26-8741*, for more information on IMS commands.

9.4.5 Issue DBRC Commands through Batch

DBRC commands were initially designed to be run as control statements for a batch utility. The commands are therefore quoted in their batch form in all manuals and literature. An example of a batch job to perform the same function as the /RML DBRC=′RECON STATUS′ command is shown in Figure 82 on page 153 with its output:

```
//DBRC    EXEC PGM=DSPURX00
//STEPLIB  DD  DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD  DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//PROCLIB  DD  DSN=IMS.SJIMSD.PROCLIB,DISP=SHR
//JCLOUT   DD  SYSOUT=(A,INTRDR)
//JCLPDS   DD  DSN=IMS.SJIMSD.JCLLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DSN=IMS.SJIMSD.PROCLIB(IMSDCLS),DISP=SHR
//*

         IMS/ESA VERSION 6 RELEASE 1  DATA BASE RECOVERY CONTROL      PA
 LIST.RECON STATUS
98.119 23:47:44.5                   LISTING OF RECON                  PA
-------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS/ESA V6R1
  DMB#=0                              INIT TOKEN=98111F2015518F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3490      DASD UNIT=3390      TRACEOFF   SSID=IMSD
  LIST DLOG=NO                 CA/IC/LOG DATA SETS CATALOGED=YES
  LOG RETENTION PERIOD=00.001 00:00:00.0

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY


 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMS.SJIMSD.RECON1
  RECON2        COPY2          IMS.SJIMSD.RECON2
  RECON3        SPARE          IMS.SJIMSD.RECON3


DSP0180I  NUMBER OF RECORDS LISTED IS        1
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 98.119 23:47:46.4
         IMS/ESA VERSION 6 RELEASE 1  DATA BASE RECOVERY CONTROL      PA
DSP0211I  COMMAND PROCESSING COMPLETE
DSP0211I  HIGHEST CONDITION CODE = 00
```

Figure 82. DBRC Commands Batch JCL and Output

A list of the valid DBRC commands is given in Appendix H, "Available DBRC Commands for DBCTL" on page 291.

9.4.6 Issue Commands from the ICMD Interface in Batch Program

You can write programs in REXX, PL1, COBOL or assembler to call IMS commands using the ICMD / RCMD interface, and to retrieve the responses. These programs can issue all IMS commands available to a DBCTL region except restart (/ERE, /NRE), shutdown (/CHE) and /MODIFY commands. This design is primarily intended to give users a means to write automated operating programs. The programs run as IMS BMPs, so you cannot write one program that combines CICS and IMS commands.

9.5 Connecting CICS TS to IMS/ESA DBCTL

To allow CICS to connect to IMS, you must have coded a DRA module, which is included in the start-up PLT. An example is shown in Figure 83.

```
   DFSPRP  DSECT=NO,
           DSNAME=IMS.SJIMSD.RESLIB,
           DDNAME=CCTLDD,
           DBCTLID=IMSD,
           MINTHRD=003,
           MAXTHRD=099,
           SOD=3,
           AGN=BMP01


Link-edit statements are thus:

//L.SYSLIN DD
//         DD *
        NAME DFSPZP0D(R)
/*
```

Figure 83. Source Code of Default DRA Module for CICS TS

This contains the specification of the IMSID to which we connect, here IMSD, and the number of threads established when we first connect (MINTHRD=003). The DRA suffix comes from the LINK-EDIT stage, here — 0D from the NAME DFSPZP0D statement. It is not specified as a macro parameter. The assembly and linking of this are illustrated in Figure 130 on page 248.

9.5.1 Automatic Connection

If you want the connection to be made automatically when CICS is started up (which is usually sensible), then

- The start-up PLT also needs to contain the entry:

  ```
  DFHPLT TYPE=ENTRY,PROGRAM=DFHDBCON
  ```

- The SIT or its overrides should contain:

  ```
  INITPARM=(DFHDBCON='0D,IMSD')
  DBCTLCON=YES
  ```

  where the IMSD is the IMSID and 0D is the suffix.

The connection is maintained until one of the following occurs:

- IMS DBCTL is closed.
- CICS is closed.
- An operator command is issued to close the connection.

The connection is then reestablished by issuing a command to restart the connection, or by restarting CICS.

9.5.2 Manual Disconnection

Issue the CDBC command from a CICS terminal to display the screen shown in Figure 84.

```
CDBC                 CICS-DBCTL CONNECTION/DISCONNECTION


          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==>
          Startup Table Suffix ==> 0D
          DBCTL ID Override    ==>



  Status of the Interface: DFHDB8293I DBCTL connected and ready.
              CICS APPLID: SCSCPAAD
```

Figure 84. CDBC Command to Disconnect CICS from IMS DBCTL

Select Option 2 to close the connection, and Option 3 to break it immediately. IMS DBCTL will not close, and treats any work in progress as an abended transaction and backs it out. CICS also performs dynamic transaction backout (DTB) on the transaction if logging is installed. The messages below are displayed:

```
DFHDB8209D DBCTL orderly disconnection requested. Press PF5 to confirm.

DFHDB8211D Orderly disconnection from DBCTL is proceeding. Check CDBC TD Q
```

9.5.3 Manual Connection

To connect CICS to DBCTL manually, issue the CDBC command from a CICS terminal to obtain the screen shown in Figure 85 on page 156.

```
CDBC                 CICS-DBCTL CONNECTION/DISCONNECTION


          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==>
          Startup Table Suffix ==> OD
          DBCTL ID Override    ==>



  Status of the Interface: DFHDB8290I DBCTL not connected to CICS.
              CICS APPLID: SCSCPAAD
```

Figure 85. CDBC Command to Connect CICS to IMS DBCTL

Then enter Option 1 to connect, and you can also change the IMSID to a different DBCTL — for example, to the ID of a TEST DBCTL, or change the DRA suffix. When you type this and press Enter, you will get the response:

```
DFHDB8207D Connection to DBCTL requested. Press PF5 to confirm.
```

When you press PF5, user should see the screen shown in Figure 86.

```
CDBC                 CICS-DBCTL CONNECTION/DISCONNECTION


          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==>
          Startup Table Suffix ==> OD
          DBCTL ID Override    ==>



  Status of the Interface: DFHDB8292I DBCTL connect phase 2 in progress
              CICS APPLID: SCSCPAAD
                 DBCTL ID: IMSD
     Startup Table Suffix: OD


PF1 = Help   2 = Refresh   3 = End
DFHDB8210D Connection to DBCTL is proceeding. Check CDBC TD queue.
```

Figure 86. Screen Showing Connect Phase 2

Press PF3 and CLEAR to exit this screen.

You can also use the command CDBI to inquire as to the status of the connection. The display should look like that in Figure 87 on page 157.

```
CDBI                      CICS-DBCTL INTERFACE INQUIRY




            Status     : DFHDB8293I DBCTL connected and ready.
            CICS APPLID: SCSCPAAD
            DBCTL ID   : IMSD
```

Figure 87. CDBI to Inquire as to the Status of the CICS to IMS DBCTL Connection

9.6 Restart

One of the major advantages of DBCTL is the ability to start and close CICS and IMS independently. However, we need to understand how each type of start and close affects both components. Table 15 on page 158 through Table 16 on page 163 show how each operation affects both sides.

Table 15 (Page 1 of 5). Restart Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS**
Type of Start How implemented	**CICS INITIAL** START=INITIAL in SIT overrides at start-up		
IMS notes threads started.	• Resource definitions loaded only from CSD and tables. • Log streams are purged. • New log token created. • Global catalog and old logs ignored. • Global catalog like the VSE restart data set. • All data from previous runs ignored.	• Resource definitions are loaded from CSD only. • Log streams are purged. • New log token is created. Global catalog and old logs ignored.	• Links started from scratch. • Previous data discarded.
Type of Start How implemented	**CICS Cold** START=Cold in SIT overrides at start-up		
• IMS notes threads started. • CICS connects to IMS specified in the DRA.	• Resource definitions loaded only from CSD and tables. • Information in the log stream for local resources is ignored, but that for remotely initiated resources is retained, using information in log stream and the global catalog. • Temporary storage queues are preserved.	Not allowed.	• Links started, and resynchronization information is preserved. • UOWs that are in doubt can be recovered from information in restarted systems.

Table 15 (Page 2 of 5). Restart Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS**
Type of Start How implemented	**CICS WARM** START=Warm in SIT overrides at start-up.		
IMS notes threads started.	• Resource definitions loaded from global catalog. • Information is not taken from the CSD and CICS tables. • All resource definitions installed from the CSD, CICS tables, installed by EXEC CICS INSTALL or CECI INSTALL or AUTOINSTALL (programs only if SIT parameter PGAICTLG=NO) or from global catalog are reinstalled from the global catalog. • The status of resources (such as out-of-service) is also remembered.	Not allowed.	• Links are restarted. • Resynchronization information is preserved. • In-doubt UOWs can be recovered from information in restarted systems.
Type of Start How implemented	**CICS EMERGENCY** START=Emergency in SIT overrides at start-up.		
• IMS notes threads started, and uses its logs to coordinate any recovery required. • CICS connects to the same IMS DBCTL it was connected to at closedown if user has installed automatic connection (DBCTLCON=YES in SIT overrides or DFHDBCON in the start-up PLT).	• Resource definitions loaded from global catalog. • Information is not taken from the CSD and CICS tables. • All resource definitions installed from the CSD, CICS tables, installed by EXEC CICS INSTALL or CECI INSTALL or AUTOINSTALL (programs only if SIT parameter PGAICTLG=NO) or from global catalog are reinstalled from the global catalog. • The status of resources (such as out-of-service) is also remembered. • Units of work that were in-flight at the last close are backed out using information in the logs and global catalog.	Not allowed.	• Links are restarted. Resynchronization information is preserved. • In-doubt UOWs can be recovered from information in restarted systems, if connection is LU 6.2. • If connection is LU 6.1, then the UOW is not rolled back if the mirror transaction in the remote system had completed when abend occurred.

Table 15 (Page 3 of 5). Restart Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS**
Type of Start How implemented	**CICS AUTO** START=AUTO in SIT overrides at start-up.		
• IMS acts depending on the option which CICS decides. • CICS will connect to the same IMS DBCTL it was connected to at closedown if user has installed automatic connection (DBCTLCON=YES in SIT overrides or DFHDBCON in the start-up PLT).	• CICS decides whether to use Warm restart if the global catalog indicates that all CICS UOWs were closed cleanly. • Otherwise CICS starts with an emergency restart. • START=AUTO is the recommended way to start CICS except when installing a new start-up list from the CSSD.	Not allowed.	• Links are restarted. • Resynchronization information is preserved. • In-doubt UOWs can be recovered from information in restarted systems, if connection is LU 6.2. • If connection is LU 6.1, then the UOW is not rolled back if the mirror transaction in remote system had completed when abend occurred.
Type of Start How implemented	**IMS INITIAL** /NRE CHECKPOINT 0 FORMAT ALL. entered as the restart command.		
• IMS starts its logging at the beginning of OLDS number 00. • IMS DBCTL formats its WADS data sets (equivalent to CICS dynamic log). • Resource definitions are loaded from MODBLKS data set, with their status as defined in the IMSGEN. • Old work units are ignored.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.

Table 15 (Page 4 of 5). Restart Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	Effect on CICS/TS (with logging)	Effect on CICS/TS (without logging)	Effect on other CICS
Type of Start How implemented	**IMS Cold** /NRE CHECKPOINT 0. entered as the restart command.		
• IMS starts its logging at the beginning of OLDS number 00. • Resource definitions are loaded from MODBLKS data set, with their status as defined in the IMSGEN. • Old work units are ignored.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.
Type of Start How implemented	**IMS Warm** /NRE. entered as the restart command.		
• IMS starts its logging at the beginning of next OLDS. • Archive job is submitted for the OLDS which was in use when IMS closed. • Resource definitions are loaded from MODBLKS data set, with their status as stored in the restart data set (RDS). • Old work units must be complete or restart will fail.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.
Type of Start How implemented	**IMS EMERGENCY** /ERE. entered as restart command.		
• IMS reads its OLDS, and backs out any work units that were in-flight when IMS closed. • Archive job is submitted for the OLDS that was in use when IMS closed. • Resource definitions are loaded from MODBLKS data set, with their status as stored in the restart data set (RDS).	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.

Table 15 (Page 5 of 5). Restart Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	Effect on CICS/TS (with logging)	Effect on CICS/TS (without logging)	Effect on other CICS
Type of Start How implemented	**IMS EMERGENCY COLD** /ERE COLDSYS. or /ERE COLDBASE. entered as restart command.		
• IMS reads the OLDS. • It does not back out any work units. This is very similar to /NRE CHECKPOINT 0 for DBCTL. • Resource definitions are loaded from the MODBLKS data set, with their status as defined in the IMS gen. • This method should be used when an /ERE. command has already failed. • Database recovery must be done manually.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.
Type of Start How implemented	**IMS AUTO** AUTO=Y in IMS start-up parameters.		
• IMS action depends on the state of the restart data sets. • If the restart data set indicates that all UOWs were completed or backed out, then IMS performs a warm start without a restart command being entered. • If the restart data set indicates that some UOWs were left in doubt, then IMS performs an emergency start without a restart command being entered.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Must use the CDBI transaction to connect CICS to IMS if CICS is up when IMS is started.	Not affected.

Table 16 (Page 1 of 5). Close Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS systems.**
Type of Close How implemented	**CICS clean close** CEMT PERF SHUT		
IMS completes outstanding UOWs and informs CICS when they are complete.	• CICS completes all outstanding tasks, closes the associated terminal sessions, and then closes. • Restart can be Warm.	CICS completes all outstanding tasks, closes the associated terminal sessions, and then closes.	• CICS completes all outstanding tasks, closes the associated MRO or ISC sessions, and then inform all users that CICS shut itself down. • Links are closed cleanly.
Type of Close How implemented	**CICS clean quickly** CEMT PERF SHUT IMM		
CICS threads in IMS that do not complete are backed out by IMS using its OLDS.	CICS purges its tasks, backing them out using the dynamic log, and then closes. Restart can be Warm.	CICS purges its tasks, backing them out using the dynamic log, and then closes.	• If remote task is running, CICS is informed of the purge and backs the task out using its own dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS purges its task after that, the remote CICS′s update is preserved, but the closing CICS backs out. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.
Type of Close How implemented	**CICS purges a task** CEMT S TASK(nnnnn) pur		
CICS threads in IMS for the task are backed out by IMS using its OLDS.	CICS backs out the task using the dynamic log. Next restart can be Warm.	CICS backs out the task using the dynamic log.	• If remote task is running, CICS is informed of the purge and backs the task out using its own dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS purges its task after that, the remote CICS′s update is preserved, but the closing CICS backs out. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.

Table 16 (Page 2 of 5). Close Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS systems.**
Type of Close How implemented	**CICS transaction abends** It happens.		
CICS threads in IMS for the task are backed out by IMS using its OLDS.	• CICS backs out the task using the dynamic log. • Next restart can be Warm.	CICS backs out the task using the dynamic log.	• If remote task is running, CICS is informed of the purge and backs the task out using its own dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS purges its task after that, the remote CICS′s update is preserved, but the closing CICS backs out. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.
Type of Close How implemented	**CICS abends, or is cancelled** It happens. (Cancel is MVS command CANCEL xxxxx from system console)		
CICS threads in IMS for the task are backed out by IMS using its OLDS.	• CICS stops, failing to write a close record to its global catalog. • Restart must be Emergency.	• CICS stops, failing to write a close record to its global catalog. • Restart must be Initial, following manual recovery of files if necessary.	• If remote tasks are running, CICS is informed of the loss of the calling task, and backs out any remote tasks using its dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS task did not, the remote CICS′s update is preserved, but the closing CICS backs out at restart. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks. • The link is not closed cleanly and may not restart easily.

Table 16 (Page 3 of 5). Close Actions for IMS DBCTL and CICS TS			
Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS systems.**
Type of Close How implemented	**IMS clean close** /CHE PURGE.		
• IMS completes outstanding UOWs and informs CICS when they are complete. It then terminates the CICS threads, and any BMPs. • Restart can be warm. • This is the same as /CHE FREEZE unless the IMS is also in transaction manager (TM) environment; in that case, the message queues are all processed before IMS closes.	CICS marks the transaction as complete in its logs.	CICS deletes all records for this transaction from its dynamic log.	Remote CICS completes all outstanding tasks.
Type of Close How implemented	**IMS closes quickly** /CHE FREEZE.		
• IMS completes outstanding UOWs and informs CICS when they are complete. It then terminates the CICS threads, and any BMPs. • Restart can be warm. • This is the same as /CHE FREEZE unless the IMS is also in TM environment; in that case the message queues are stopped, and no new messages are processed.	CICS marks the transaction as complete in its logs.	CICS deletes all records for this transaction from its dynamic log.	Remote CICS completes all outstanding tasks.

Table 16 (Page 4 of 5). Close Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	Effect on CICS/TS (with logging)	Effect on CICS/TS (without logging)	Effect on other CICS systems.
Type of Close How implemented	**IMS purges CICS thread** /STO REG nn. , or /STO REG ABDUMP.		
• IMS stops the unit of work and backs out any updates for it using its OLDS. It then informs CICS of the failure. • Next restart of IMS can be Warm.	• CICS backs out the task using the dynamic log. • Next restart can be Warm.	CICS backs out the task using the dynamic log.	• If remote task is running, CICS is informed of the purge and backs the task out using its own dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS purges its task after that, the remote CICS′s update is preserved, but the closing CICS backs out. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.
Type of Close How implemented	**CICS thread abends** It happens.		
• CICS thread in IMS is backed out by IMS using its OLDS. • Next restart can be warm.	• CICS backs out the task using the dynamic log. • Next restart can be Warm.	CICS backs out the task using the dynamic log.	• If remote task is running, CICS is informed of the purge and backs the task out using its own dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS purges its task after that, the remote CICS′s update is preserved, but the closing CICS backs out. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.

Table 16 (Page 5 of 5). Close Actions for IMS DBCTL and CICS TS

Effect on IMS/ESA	**Effect on CICS/TS (with logging)**	**Effect on CICS/TS (without logging)**	**Effect on other CICS systems.**
Type of Close How implemented	**IMS abends, or cancelled** It happens. (Cancel is MVS command CANCEL xxxxx from system console)		
• All CICS threads in IMS for the task fail. • They are backed out in IMS using its OLDS at emergency restart.	• CICS abends all tasks which had IMS work in progress using the dynamic log. • Restart can be warm. • CICS will connect to IMS only when it restarts by the CDBI command.	• CICS abends all tasks which had IMS work in progress using the dynamic log. • Restart can be warm. • CICS will connect to IMS only when it restarts by the CDBI command.	• If remote tasks are running, CICS is informed of the loss of the calling task, and backs out any remote tasks using its dynamic log. • If link is LU 6.1, then if a mirror task has completed but the closing CICS task did not, the remote CICS′s update is preserved, but the closing CICS backs out at restart. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.
Type of Close How implemented	**OS/390 abends** It happens occasionally. (For example, power failure).		
• IMS must be restarted with an emergency restart. • IMS then backs out all in-flight transactions using its OLDS. • Connection between CICS and IMS may need to be restarted with the CDBI transaction.	• CICS must be restarted with an emergency restart. • CICS will back out all in-flight transactions using its log.	• CICS is restarted with an INITIAL start. • Manual recovery will be necessary on any files that were being updated at the time of failure.	• If remote tasks were running, CICS is informed of the loss of the calling task, and backs out any remote tasks using its dynamic log. • This will be done with an emergency restart if the remote CICS system was also on the failing MVS image. • If link is LU 6.1, then if a mirror task has completed but the closing CICS task did not, the remote CICS′s update is preserved, but the closing CICS backs out at restart. • If the link is LU 6.2, then both CICS coordinate the back-out of their tasks.

9.7 Logging

Users now will have two logs to manage. Although the CICS has been moved to MVS, the environment is very similar to the VSE environment. The IMS DBCTL logs need a great deal of attention.

9.7.1 IMS Logging

There are two logs in IMS DBCTL, the write-ahead data set (WADS) and the online log data set (OLDS). The following sections describe their functions and use.

9.7.1.1 WADS

IMS logs all its updates to its WADS and to its OLDS. The WADS are designed to be written very fast, and are seldom read. IMS writes its database buffers to the WADS. The real database files are then updated by separate subtasks from the buffers. Should this updating fail for some reason, then the updates can be reapplied from the WADS — for example, at restart.

WADS are normally duplexed, but this is optional. Users would also normally define some spares, but this is also optional.

You define whether you want dual or single logging with the IMS start-up parameter WADS=D or =S. We defined it in IMS.SJIMSD.PROCLIB(DSFPBSJD) for our project, and defined the number of WADS overall in IMS.SJIMSD.PROCLIB(DFSVSAMB) with WADSDEF WADS=(0,1,2). This defines the number, and the suffixes they should have.

9.7.1.2 OLDS

IMS also logs its updates, before images, checkpoints, messages, statistics and trace dumps, along with any record issued by the /LOG command to the OLDS. These are normally dual (a primary and a secondary) but the secondary is optional, as set in the IMS start-up parameters in the member IMS.SJIMSD.PROCLIB(DFSPBSJD). We used this:

```
OLDSDEF OLDS=(00,01,02,03,04),BUFNO=005,MODE=DUAL.
```

The statement above gives the number of OLDS and whether they are single (names OLPnn) or dual (names OLPnn and OLSnn).

When a pair of OLDS becomes full, IMS switches to another pair automatically, and submits a job to archive them to an SLDS. The OLDS is also used for on-line back out of IMS database updates in a failing CICS thread or BMP region. IMS will not reuse an OLDS until it has been archived successfully, and it no longer has any incomplete work units in it.

A long-running BMP that does not issue checkpoints could therefore cause all the OLDS to be unavailable in a busy system. For this reason, you can add extra OLDS pairs to the start-up set simply by following these steps:

1. Code and assemble/link a set of dynamic allocation macros for extra OLDS into USERLIB. See B.6, "Define Dynamic Allocations for IMS" on page 245.
2. Allocate the appropriate data sets. The block-end LRECL size should match your existing OLDS. The size can be whatever you like.

3. Issue an IMS start command /STA OLDS nn. to tell IMS to use the OLDS pair.

4. Issue an IMS stop command /STO OLDS nn. to remove the OLDS pair from IMS when you no longer need them.

For example

1. Code the following source:

   ```
   &SYS1  SETC 'SJIMSD'
   DFSMDA TYPE=INITIAL
   DFSMDA TYPE=OLDS,DDNAME=DFSOLS69,DISP=SHR,DSNAME=IMS.&SYS1..OLS69
   DFSMDA TYPE=OLDS,DDNAME=DFSOLP69,DISP=SHR,DSNAME=IMS.&SYS1..OLP69
   DFSMDA TYPE=FINAL
          END
   ```

   Assemble and link-edit it to USERLIB.

2. Allocate the data sets IMS.SJIMSD.OLP69 and IMS.SJIMSD.OLS69 with DBC of RECFM=VB,LRECL=18428,BLKSZE=18432.

3. Issue this command, when needed, through the system console:

```
IMSDSTA OLDS 69.
```

The OLDS pair 69 is added and will be used by the IMS DBCTL and will remain allocated to IMS DBCTL until either IMS is closed or an operator issues the console command

```
IMSDSTO OLDS 69.
```

9.7.1.3 Log Archiving

The OLDS are archived, when they are full, by a job that IMS submits automatically using DBRC. It is often named ARCIMSD and is stored in DBRC symbolic skeleton form in some model JCL library. This is IMS.SJIMSD.JCLLIB(ARCHJCL) in our case.

An example of an archive job is shown in Figure 141 on page 264. This job archives the primary OLDS to an archive file on disk DFSSLOGP whose name includes a date and time stamp. If you include a DD card for DFSSLOGS, it can also write a second copy.

9.7.1.4 SLDS

DBRC keeps a record of the name and volume ID of all SLDSP and SLDSS data sets created even after they have been deleted from the catalog. There are several options you can choose from for storing the SLDSs:

- Keep one SLDSP set on disk, for forward or backward recovery for the same number of days as users keep the database backups. Use SMS to delete them after this time.

- Keep one SLDSP set on disk, for forward or backward recovery. Use SMS to archive them to tape within 24 hours of creation, and delete them after the same number of days as users keep database backups.

- Keep two SLDS sets on disk: one set of SLDSPs and one of SLDSS. Use these for forward or backward recovery. Keep both for the same number of days as users keep the database backups. Use SMS to archive the SLDSP to tape within 24 hours of creation, and archive the SLDSS to tape immediately. Use SMS to delete them after the same number of days as you keep database backups.

- Write SLDSP data sets directly to tape, which the user deletes from the MVS catalog within 5 days.
- Write SLDSP and SLDSS data sets directly to tape, which the deletes them from the MVS catalog within 5 days. Keep the SLDSP tapes on-site. Keep the SLDSS tapes off-site in a fire-proof tape disaster store.

9.7.1.5 CICS Journals

CICS writes its information to the MVS logger. MVS copies this off to CICS secondary log data sets on disk. You can then copy these off to tertiary logs on disk or tape as your users need.

9.8 Database Backup and Recovery

Within this section, we describe basic database backup and recovery operation using IMS production utility programs. In Chapter 10, "Backup and Restore Using DB Tools" on page 175, we introduce some DB tools that make operation a lot easier.

9.8.1 Image Copy

Users should employ IMS image copy to back database up periodically. This can either be run with the database online using the operator command from CICS:

```
/RMG DBRC='IC DBD(dbd name)'
```

This causes DBRC to generate a job to run an image copy of a database. The JCL is held in DBRC symbolic form in member IMS.SJIMSD.JCLLIB(ICJCL). An example of a batch DBRC job is shown in Figure 142 on page 265.

You can also issue the same DBRC command in batch job format:

```
//DBRC     EXEC PGM=DSPURX00
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//PROCLIB  DD  DSN=IMS.SJIMSD.PROCLIB,DISP=SHR
//JCLOUT   DD  SYSOUT=(A,INTRDR)
//JCLPDS   DD  DSN=IMS.SJIMSD.JCLLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  GENJCL.IC DBD(dbd name)
//
```

This has the same effect if IMS is not running. All the above methods lead to creation and submission of an image copy batch job. You can keep the JCL and run the same batch job directly. An example of an image copy job is shown in Figure 144 on page 268.

9.8.2 Database Recovery

To recover a database, use the IMS database recovery utility. This can be run with the database online using the operator command from CICS:

```
/RMG DBRC='RECOVER DBD(dbd name)'
```

This causes DBRC to generate a job to recover the database. The JCL is held in DBRC symbolic form in a member of the JCL model library. The member is IMS.SJIMSD.JCLLIB(RECOVJCL) in our case. An example of a batch DBRC job is shown in Figure 143 on page 266

You can also run a batch job with the embedded DBRC command:

```
GENJCL.RECOVERY DBD(dbd name)
```

This has the same effect if IMS is not running.

The advantage of getting DBRC to generate the job is that since DBRC keeps track of the OLDS and SLDS data sets to which updates to each database have been logged, it can also supply those names to the JCL through the skeleton.

You can also run the recovery from a batch job with the user's own code. You, therefore, need to ensure that you have named all relevant SLDS data sets to perform forward recovery. If you miss some, the resultant database will be inconsistent.

If all you want is to restore the image copy, and not recover it forward, then you can set DFSULOG to DD DUMMY instead. An example of an image copy job is shown in Figure 145 on page 269.

9.9 Application Operation

Since only a part of the migration work has been done, we cannot show the complete production operation. Members of the project team have returned to their homeland where they continue to carry out the work. The following is a brief description.

9.9.1 Online Operation

Although there is newer technology that can be used to make the system easier to operate or make it look better. We decided not to use it. Our goal has been to keep as much transparency as possible, to minimize the reeducation needed by the user community. Therefore, the online screens look exactly as they did in the VSE environment. For example, log on to CICS TS and type:

```
CZFY
```

Press Enter, and the screen shown in Figure 88 on page 172 appears.

```
CZFYHX2                 SAFE II 3270 TRANSACTION DATA

-----------------------------------------------------------------------
TRNCD 6771     CTLID 0000     WSID 0000      TRNID 00000000
 USER 200      USERA 0        AUTH 00        TRIND 00
FIELD 1 20110101234        FIELD 2  4
FIELD 3 12345              FIELD 4  1
FIELD 5 20110101234        FIELD 6  0
FIELD 7 0                  FIELD 8  0
FIELD 9 0                  FIELD10  0
FIELD11 0                  FIELD12  0
TOBFMH  030009  TOBAUTH  09  TOBINDS  00  TOBTSER  000000000000
020500FF F0F0F8F4 F6F571FF             *....008465..    *







-----------------------------------------------------------------------

 F1=Help    F3=Exit    F5=Rfind  F12=Cancel
```

Figure 88. Sample of First SAFEII Response Screen

When you input accounting information by overtyping the displayed fields and press Enter again, another screen appears (Figure 89).

```
CZFYHX2                 SAFE II 3270 TRANSACTION DATA

----------------------------------------------------------------------
TRNCD 6734     CTLID 0000     WSID 0000      TRNID 00000000
 USER 200      USERA 0        AUTH 00        TRIND 00
FIELD  1 20110101234         FIELD 2  0
FIELD 3 0                  FIELD 4  0
FIELD 5 0                  FIELD 6  0
FIELD 7 0                  FIELD 8  0
FIELD 9 0                  FIELD10  0
FIELD11 0                  FIELD12  0
TOBFMH  034001  TOBAUTH  00  TOBINDS  00  TOBTSER  000000000000
0C340402 3400010E 28C1405B 6A4DC40F    *.........A $.(D.*
28C140F2 F0F1F1F0 F1F0F1F2 F3F44040    *.A 20110101234  *
0E28C140 4EBD4EC5 58420F28 C1404040    *..A +.+E....A   *
150E28C1 404EC558 425B6A4D C40F28C1    *...A +E..$.(D..A*
40F2F0F1 F1F0F1F0 F1F2F3F4 400E28C1    * 20110101234 ..A*
4050FA51 C90F28C1 40404040 4040F14B    * &..I..A      1.*
F2F3F4F5 150E28C1 404F894C 5D4E9E55    *2345...A |.<)+..*
FC0F28C1 40404040 40404040 40404040    *...A            *
40404040 F04BF0F0 150E28C1 404AFA4C    *    0.00...A ¢.<*
-------------------------------------------------------------------
 F1=Help    F3=Exit    F5=Rfind  F12=Cancel
```

Figure 89. Sample of Second SAFE II Response Screen

9.9.2 Batch Operation

All VSE production JCL is converted to MVS. Addition or deletion takes place to reflect the particular operation environment.

For long runs with multiple-step batch jobs, we also plan to insert conditional and restartable steps, so that jobs can be restarted in midprocess in case an abend condition occurs.

Chapter 10. Backup and Restore Using DB Tools

To those users who are new to the OS/390 environment, JCL coding is one of the more time-consuming learning exercises. We, therefore, introduce an IMS tool that can make operation much easier. In general, IMS DB tools perform better than product-supplied utilities. However, our primary goal is ease of use. A few new tools, just recently announced, have an easy ISPF front-end interface. We next introduce a tool, called *IMS Database Control Suite*, that supports our goal.

10.1 IMS Database Control Suite

IMS Database Control Suite (hereafter called *Control Suite*) provides an interface to the IMS tools and utilities needed to manage and service IMS databases. Control Suite allows users to quickly setup processes for the day-to-day and exception management of databases.

The DB Control Suite provides these facilities:

- An ISPF front-end to:
  - Back up databases.
  - Recover databases.
  - Monitor and reorganize databases.
  - Manage IMS libraries.
  - Perform miscellaneous utility functions.
- Interfaces to the IMS base utilities, IBM IMS-enhanced utilities, and third-party vendor utilities.
- On-demand database space monitoring.
- An easy-to-use setup operation to establish database groups or applications to be managed and to generate appropriate JCL.
- An easy-to-use interface to DBRC.
- Other library management utilities.

Control Suite can be configured to manage all or some of the databases for an IMS system. It is made up of Control Suite groups, which are related databases, database applications, or other groupings determined by the users. The groups are defined as part of the setup process, where all needed database information is collected, and JCL is built for future use to back up, recover, monitor, and reorganize databases and perform RECON maintenance.

During the setup process, you select the products for which JCL is to be generated. For example, users may want to use a non-IBM database unload product. When you select the appropriate option, JCL for this product is prebuilt for subsequent execution. All JCL generated is optimized for performance where possible.

The Control Suite should be able to offer users a new, higher level of control over the day-to-day and exception management of databases. Typically, database management requires extensive IMS skills and in-depth understanding of the IMS DBRC procedures. Control Suite provides a unique platform that interfaces to these processes, saving both time and effort, especially to those who are new to OS/390 or are migrating to an IMS DBRC-controlled environment.

The main benefit of Control Suite is that it allows DBAs to perform their daily database support activities in a timely and efficient manner, even with minimal knowledge of IMS DB, DBRC, and IMS utilities.

Control Suite lets the DBA:

- Avoid input data redundancy. Any time there is a change in the IMS DB environment, such as the addition of a new database, changes in naming conventions, data set definitions, or in storage allocations, many JCL and control statements need to be modified.

  Control Suite greatly simplifies this task. The input to the changes is specified in a single, easily maintainable data set, and the required JCL maintenance is performed automatically.

- Minimize data entry. Only data that cannot be extracted by the system from other sources, such as DBD, DDN, data set names, space allocations and some processing options, must be specified manually. Other required data, such as database organization, logical relationships, and record and key lengths, is extracted from the IMS DBD library automatically.
- Benefit from full flexibility for naming conventions. Although Control Suite requires certain standards regarding the naming of its own data sets, there is no restriction in the use of any naming structure.
- Automate the generation of JCL for IMS and IBM tools and other supplier utilities. In the past, the JCL for major utilities, such as database reorganization, database timestamp recovery, database batch backout, and DBT or OEM pointer checker utility, had to be set up manually. This required advanced skills and therefore was subject to errors. Control Suite provides an automated way to generate this JCL, reducing the chance of errors.
- Easily create and maintain JCL skeletons.
- Use an interactive front end for the inquiry and maintenance of the RECON data set. In DBRC, inquiry against the RECON data set is performed by issuing the LIST command. The resulting printed output must then be scanned for the appropriate data. Commands for RECON maintenance must be set up manually and incorporated into a batch job for execution. Finding the association between the various RECON records for a specific database also requires advanced skills.

The Control Suite also assists users with monitoring and database maintenance functions that are not given in detail here. Refer to the product manual for more information. In summary, Control Suite does the following:

- Provides an easy-to-use tool to collect and build the data required by the JCL generation process.
- Generates JCL to support the standard IMS database utilities, IMS database tools (DBT), vendor utilities, and IMS log utilities.
- Generates JCL for database backup utilities.
- Generates JCL for the IMS database forward or timestamp recovery process.
- Displays and updates the contents of the DBRC RECON data sets using an ISPF front end.

- Provides library databases maintenance functions.

Control Suite contains all the tools necessary to maintain and manage the IMS databases from a central focus point. It also provides an open interface that allows tools supplied by others than IBM to be easily integrated.

10.2 SAFEII Application

We installed the Control Suite during the residency project and tried it out on generating the complete database backup, restore, and recovery processing. The following is a record of our exercise.

We execute the Control Suite under the ISPF command shell (Option 6) environment. For more permanent installation, it can be integrated into applications under ISPF. There are many way to customize the installation which we do not get into. The shell command looks like:

```
                                 ISPF Command Shell
Enter TSO or Workstation commands below:

===> ex 'dbt.imsdcs.sidlcexe(idlc00)'  'dbt.imsdcs  dbt.imsdcs'
```

Here, we use DBT.IMSDCS as our Control Suite system high level qualifier. When you press Enter, the software logo is displayed, as shown in Figure 90, and is followed by the appearance of a manual application selection screen. In general, the application should already exist and should have all environmental files built.

```
Press ENTER to continue.

              =======   ============     ======      ======  *
              =======   ==============   =======    =======
                ===       ===    =====     ======   ======
                ===       ===========      ======= =======
                ===       ===========      === ======= ===
                ===       ===     ====     ===  =====  ===
              =======   ==============   =====   ===   =====
              =======   ============     =====    =     =====

         Information Management System IMS Database Control Suite
                    (IMS DB CS)  Version 1  Release 1
               ---------------------------------------------
               | Licensed Materials - Property of IBM      |
               | 5697-D15 (C) Copyright IBM Corp. 1998     |
               | All Rights Reserved.                      |
               | US Government Users Restricted Rights -   |
               | Use, duplication or disclosure restricted |
               | by GSA ADP Schedule Contract with IBM Corp.|
               ---------------------------------------------
 * Trademark of International Business Machines


 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 90. Database Control Suite Logo Screen

Control Suite displays messages and starts to build environmental information files for the new application. Control Suite can handle multiple applications, but it works on one application at a time. The message displayed after you enter an application selection looks like that in Figure 91 on page 178.

```
                            IMS DB CONTROL SUITE

Command ===>

Press END to exit.

Control Suite ID List

New Control Suite ID . . IMSD

Select(S), view users(V) or delete(D)
  CS ID   IMS VERSION  IN USE
****************************** Bottom of data ************************




------------------------------------------------û
| IMSD not defined to IMS DB Control Suite      |
| Press ENTER to initiate setup or END to exit. |
-------------------------------------------------   F5=RFIND     F6=R CHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT    F12=RETRIEVE
```

Figure 91. Message Responding to Selection of IMSD

The Control Suite main panel (Figure 92 on page 179) is then displayed. Because application IMSD is an application new to Control Suite, and no information has yet been collected, so Control Suite does not yet know which version of IMS the IMSD is using..

```
                           IMS DB CONTROL SUITE
                    for CS ID IMSD     IMS Version ?
Command ===>
                                                          TIME. .14:12:18.79
                                                          DATE. .1998/08/13
                                                          JDTE. .1998.225
Select a task or press END to exit.

Task. . . .     1. Set up operating environment
                2. Backup databases
                3. Recover databases
                4. Monitor/Reorganize databases
                5. Maintain RECON
                6. Perform library maintenance
                7. Use toolbox




--------------------------------------------------------------------------
| SETUP DEFINE, COLLECT and BUILD not completed for at least 1 CS Group. |
| No tasks selectable.                                                   |
-------------------------------------------------------------------------- GE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 92. Database Control Suite Main Panel

The main panel clearly describes the functions the Control Suite offers.

10.2.1 IMS Environment Data Collection

The first thing the Control Suite needs to know is the identify of the application itself. Selecting Option 1 from the main panel, we select the environment collection phase of Control Suite operation. The subtask shown in Figure 93 are then displayed.

```
                           IMS DB CONTROL SUITE
SETUP              function for CS ID IMSD     IMS Version ?
Command ===>
                                                           TIME. .14:17:28.47
                                                           DATE. .1998/08/13
                                                           JDTE. .1998.225
Select a subtask to continue or END to exit.

Subtask . .     1. Define Control Suite environment
                2. Collect/update CS Group data
                3. Build customized CS Group libraries
                4. Display built environment
                5. Integrate CS Groups into the IMS environment
```

Figure 93. Subtask List for Option 1 of the Main Panel

The panel shown in Figure 94 on page 180 presents the data collection option. Control Suite collects this information about the OS/390 operation environment. The typical information needed includes the naming convention for the application system and Control Suite work files, whether files are managed by the OS/390 SMS subsystem, what file storage medium is used, and so on.

```
                             IMS DB CONTROL SUITE          Enter required field
SETUP    DEFINE    GLOBAL   function for CS ID IMSD     IMS Version ?
Command ===>
Press ENTER to continue or END to exit.

Permanent data sets hlq       . . DBT.IMSDCS
Work data sets hlq            . . DBT.IMSD.WORK
SORT library dsname           . . SYS1.SORTLIB

SMS                           . . N          (Y or N)
VOLSER                        . . TOTTS1     (blank if SMS=Y)
DASD device type              . . 3390       (3380 or 3390)
DASD unit                     . . SYSDA      (Esoteric name or blank if SMS=Y)
TAPE unit                     . . 3490       (Esoteric name)
Global jobs job name prefix   . . imsd       (1 to 4 characters)
Tape retention                . . 21         (Days)
Work space - Utilities        . . 20         (Primary cylinders)
                              . . 10         (Secondary cylinders)
Work space - SORT             . . 20         (Primary cylinders)
                              . . 10         (Secondary cylinders)
I/O buffers                   . . 50         (VSAM)
                              . . 50         (OSAM)
```

Figure 94. Panel Showing Entry Fields for Data Collection Option

Next, the Control Suite wants to know what type of utilities the application uses (Figure 95 on page 181). Users can select IMS product utilities, IMS DB Tools, or utilities from other vendors, such as BMC or NEON. Preference options 1 through 4 give the search order when performing JCL generation. When a special function is needed, image copy for example, if the execution load library of image copy for Preference 1 does not exist, the next one will be used. If none is found, then the default IMS product utilities are used to generate the JCL for that function.

```
                         IMS DB CONTROL SUITE
SETUP    DEFINE    UTILITY  function for CS ID IMSD     IMS Version ?
Command ===>

Press ENTER to continue or END to exit

Select IBM or Vendor(s) IMS utilities to be used in JCL generation in order of
preference. "1" is first preference and "4" is lowest preference.

Preferences
IBM                . . N         (N or preference 1 through 4)
BMC                . . N         (N or preference 1 through 4)
IDI                . . N         (N or preference 1 through 4)
NEON               . . N         (N or preference 1 through 4)
IMS supplied       . .           (Default)

IBM Tools
HPR                . . N         (Y to select High Performance Reorg)
DBT                . . N         (Y to select Database Tools)
IB                 . . N         (Y to select Index Builder)
```

Figure 95. Control Suite Preference Options

The Control Suite also needs to know the IMS online region system data sets (Figure 96).

```
                           IMS DB CONTROL SUITE
SETUP    DEFINE    IMS      function for CS ID IMSD     IMS Version 6.1
Command ===>

Press ENTER to continue or END to exit.

RECON data sets
Maxrecl. . 20
RECON1 . . IMS.SJIMSD.RECON1                          Vol => TOTTS1 Cyl => 10
RECON2 . . IMS.SJIMSD.RECON2                          Vol => TOTTS2 Cyl => 10
RECON3 . . IMS.SJIMSD.RECON3                          Vol => TOTTS3 Cyl => 10
IMS system data sets
DBDLIB . . IMS.SJIMSD.DBDLIB
PSBLIB . . IMS.SJIMSD.PSBLIB
RESLIB . . IMS.SJIMSD.RESLIB
ACBLIB . . IMS.SJIMSD.ACBLIB
EXITLIB. . >>> IMS.SJIMSD.EXITLIB
DYNLIB . . >>> IMS.SJIMSD.DYNLIB
MACLIB . . IMS.SJIMSD.MACLIB
WADS   . . >>> IMS.SJIMSD.DFSWADS                        No. of WADS . . 3
PRISLDS. . IMS.SJIMSD.SLDSP                              GDG . . N  (Y or N)
SECSLDS. . IMS.SJIMSD.SLDSS                              GDG . . N  (Y or N)
```

Figure 96. Identifying the IMS Online Data Sets

The panel shown in Figure 96 could be repeated several times until all IMS system data sets are collected. If a file is not found, Control Suite marks **> > >** in front of the file for correction. Also, once online system data set information is gathered, Control Suite analyzes it and finds out which version of IMS the online region runs under. The region is reflected on the next panel by filling in the IMS version number instead of the '?' shown on previous panels.

10.2.2 Application Data Collection

In the next phase, Control Suite builds the application detail information (see Figure 97). Database information could come from the DBD library or from the ACBLIB. It is not an error if the same DBD name shows twice on the listing.

```
SETUP    COLLECT    function for CS ID IMSD     IMS Version 6.1
Command ===>

Press ENTER to continue or END to exit.

Specify Source Options to Collect Data for CS Group . . IMSD

DB type                . . DLI
Build DBD list         . . N         (Y or N)
Batch mode             . . N         (Y or N)
Check ACBLIB           . . N         (Y or N)
DBD names list         . . ?
IMS DBDLIB library     . . IMS.SJIMSD.DBDLIB
Alternate DBD library  . .
IMS DYNLIB library     . . IMS.SJIMSD.USERLIB
User DB dsnames list   . .
```

Figure 97. Collecting Application Details for Control Suit

As soon as application information is provided, Control Suite displays an ISPF edit screen with all information gathered from DBD and ACB libraries. Users then select the real application system information by deleting all DBDs not belonging to the target application (Figure 98).

```
-     Application data collection search input
-     ========================================
-
-  Data will be collected for DBD(s) selected.
-
-  At least 1 DBD of a Prime or Index database set must be specified.
-  Delete DBD(s) not needed from list below.
-
-DBDNAME
--------
 CUSTDB
 CUSTSI
 CZFBTBE
 CZFBTBK
 .......
```

Figure 98. ISPF Edit Screen Sample

After selecting all application DBDs, the detail listing looks like Figure 99 on page 183.

```
EDIT       DBT.IMSDCS.IMSD.FDBDLIST(IMSD) - 01.00          Columns 00001
Command ===>                                                  Scroll ===
****** ***************************** Top of Data **********************
000001 -     Application data collection search input
000002 -     ========================================
000003 -
000004 -  Data will be collected for DBD(s) selected.
000005 -
000006 -  At least 1 DBD of a Prime or Index database set must be specify
000007 -  Delete DBD(s) not needed from list below.
000008 -
000009 -DBDNAME
000010 --------
000011 CZFBTBE
000012 CZFBTBK
000013 CZFBTLE
000014 CZFBTLK
000015 CZFBTOE
000016 CZFBTOK
```

Figure 99. Control Suite Detail Listing

After the selection is done, Control Suite displays a few more interactive panels for users to customize job name, database share level, and so on, for production operation. Finally, it displays the panel shown in Figure 100 to confirm the application.

```
SETUP    COLLECT    function for CS ID IMSD     IMS Version 6.1
Command ===>

Press ENTER to continue or END to exit.

Database Data Set Control Variables for CS Group . . IMSD

Sort(A or D)
  DBDNAME   DDNAME TYP ACM  CI/BLKSZ  VSAMSHR  VSAMFRSP  REPRO
Sort  :
Propagate global changes:
  CZFBTBE  CZFBTBE  P  ESDS    2048     2,3     0,0
  CZFBTBK  CZFBTBK  X  KSDS    1024     1,3     0,0       Y
  CZFBTLE  CZFBTLE  P  ESDS    2048     1,3     0,0
  CZFBTLK  CZFBTLK  X  KSDS    1024     1,3     0,0       Y
  CZFBTOE  CZFBTOE  P  ESDS    2048     1,3     0,0
  CZFBTOK  CZFBTOK  X  KSDS    1024     1,3     0,0       Y
****************************** Bottom of data *****************
```

Figure 100. Final Control Suite Data Collection Panel

10.2.3 Operation Procedures Generation

At this point, Control Suite has completed its data collection phase. The next panel is building the application operation procedures (Figure 101 on page 184).

```
SETUP    BUILD      function for CS ID IMSD     IMS Version 6.1
Command ===>

Press END to exit.

Build all             . . N  (Y or N - Build all CS Groups listed below)
Batch mode            . . N  (Y, N or C)
Alternate DBDLIB dsn  . .
Build selected jobs
LOCAL.JOBLIST member  . .         EDIT . . N  (Y or N) Process . .    (Y or N)

Select(S)
   . . IMSD       . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
   . .            . .          . .           . .           . .
```

Figure 101. Control Suite Panel for Building Application Procedures

10.2.4 Database Maintenance Operation

Once operation procedures are built, users can then perform any of the database maintenance functions, such as:

- Back up databases.
- Recover databases.
- Monitor and reorganize databases.
- Maintain RECON.
- Perform library maintenance.

as shown in Figure 92 on page 179 of the CS main panel. Control Suite builds all necessary JCL and jobs according to the function the users selected. Users can then just verify (optional) the JCL and submit the jobs. The operation is done. If user installations have other automation software installed, further automation of the database maintenance functions is also available. Figure 102 on page 185 shows a sample database backup job for the residency project generated by the Control Suite.

```
//CH01IC1   JOB (999,POK),'CHEN,S',REGION=0M,CLASS=A,
//        MSGCLASS=T,MSGLEVEL=(1,1),NOTIFY=CHENS
//*
//PROCLIB  JCLLIB ORDER=DBT.IMSDCS.IMSD.LOCAL.PROCLIB
//*
//*------------------------------
//*     VERIFY DATASET           |
//*------------------------------
//VFY001 EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//CZFBTBE  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTBE
//CZFBTLE  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTLE
//CZFBTOE  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTOE
//CZFBTBK  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTBK
//CZFBTLK  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTLK
//CZFBTOK  DD DISP=SHR,DSN=CASE.ZJLS.CZFFTOK
//SYSIN    DD *
 VERIFY FILE(CZFBTBE)
 VERIFY FILE(CZFBTLE)
 VERIFY FILE(CZFBTOE)
 VERIFY FILE(CZFBTBK)
 VERIFY FILE(CZFBTLK)
 VERIFY FILE(CZFBTOK)
/*
//*
//*---------------------------
//*  IMAGE COPY STEP          |
//*---------------------------
//IC002 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTBE   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTBE
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTBE.CZFBTBE(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(TRK,(2,2),RLSE)
//SYSIN    DD *
D1 CZFBTBE  CZFBTBE  DATAOUT1
/*
```

Figure 102 (Part 1 of 4). Sample Database Backup JCL Generated by Control Suite

```
//*
//*---------------------------
//*  IMAGE COPY STEP         |
//*---------------------------
//IC003 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTLE   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTLE
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTLE.CZFBTLE(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(CYL,(25,2),RLSE)
//SYSIN    DD *
D1 CZFBTLE  CZFBTLE  DATAOUT1
/*
//*
//*---------------------------
//*  IMAGE COPY STEP         |
//*---------------------------
//IC004 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTOE   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTOE
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTOE.CZFBTOE(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(CYL,(25,2),RLSE)
//SYSIN    DD *
D1 CZFBTOE  CZFBTOE  DATAOUT1
/*
```

Figure 102 (Part 2 of 4). Sample Database Backup JCL Generated by Control Suite

```
//*
//*---------------------------
//*  IMAGE COPY STEP          |
//*---------------------------
//IC005 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTBK   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTBK
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTBK.CZFBTBK(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(TRK,(20,2),RLSE)
//SYSIN    DD *
D1 CZFBTBK  CZFBTBK  DATAOUT1
/*
//*
//*---------------------------
//*  IMAGE COPY STEP          |
//*---------------------------
//IC006 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTLK   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTLK
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTLK.CZFBTLK(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(TRK,(3,2),RLSE)
//SYSIN    DD *
D1 CZFBTLK  CZFBTLK  DATAOUT1
/*
```

Figure 102 (Part 3 of 4). Sample Database Backup JCL Generated by Control Suite

```
//*
//*---------------------------
//*  IMAGE COPY STEP         |
//*---------------------------
//IC007 EXEC PGM=DFSUDMP0,REGION=4096K,COND=(0,LT),
//             PARM='DBRC=Y'
//STEPLIB  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//         DD DISP=SHR,DSN=IMS.SJIMSD.EXITS.RESLIB
//SYSPRINT DD SYSOUT=*
//DFSRESLB DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS      DD DSN=IMS.SJIMSD.DBDLIB,DISP=SHR
//RECON1   DD DISP=SHR,DSN=IMS.SJIMSD.RECON1
//RECON2   DD DISP=SHR,DSN=IMS.SJIMSD.RECON2
//RECON3   DD DISP=SHR,DSN=IMS.SJIMSD.RECON3
//CZFBTOK   DD DISP=SHR,AMP=('BUFND=50'),
//            DSN=CASE.ZJLS.CZFFTOK
//DATAOUT1 DD DISP=(,CATLG),
//            DSN=DBT.IMSDCS.IMSD.CZFBTOK.CZFBTOK(+1),
//            DCB=(DBT.IMSDCS.IMSD.DASD.IGDGMODL,BUFNO=50),
//            UNIT=(SYSDA),
//            SPACE=(TRK,(3,2),RLSE)
//SYSIN    DD *
D1 CZFBTOK  CZFBTOK  DATAOUT1
/*
//*************************************************************
//*   SET COND CODE
//*************************************************************
//SETCC EXEC PGM=IDLSETCC,COND=(0,LT)
//STEPLIB  DD DISP=SHR,DSN=DBT.IMSDCS.SIDLLMD0
//***************************************************************
//*   ABEND IF ERROR
//***************************************************************
//ALERT EXEC PGM=IDLABND,COND=(88,EQ,SETCC)
//STEPLIB  DD DISP=SHR,DSN=DBT.IMSDCS.SIDLLMD0
//* #EOJ#
```

Figure 102 (Part 4 of 4). Sample Database Backup JCL Generated by Control Suite

For the database forward-recovery function that requires complex time-stamped information, the Control Suite searches RECON data sets to find log information for the database and generates the necessary recovery JCL accordingly.

10.2.5 Conclusion

We find that the Control Suite is rather easy to operate. Users with simple TSO ISPF operation ability and basic application knowledge should be able to use this tool to maintain any IMS application without difficulty.

Another good feature of the Control Suite is that once all related information is collected, the Control Suite generates a list of all required database maintenance tasks, shown in Figure 103 on page 189.

```
TOOLBOX  GENJCL     function for CS ID IMSD     IMS Version 6.1
Command ===>

Press END to exit.

List of all available jobs in CS Group IMSD

Update job list  . . N        (Y or N)
Show DB grouping . .          (Group # or DBD name (full or generic))

Select(S) or Sort(A or D)
  ---- Job name ----
  CS BUILT   LOCAL     GRP  JOB DESCRIPTION
Sort:
  CH01SAJC                  ALLOCATE DATASETS
  CH01IC1              1    IMAGE COPY FOR DB'S BY GRP #
  CH01OIC1             1    ONLINE IC FOR DB'S BY GRP #
  CH01PI1              1    PTR.CHK FOR DB'S BY GRP# USING IMG CPY
  CH01PD1              1    PTR.CHK FOR DB'S BY GRP# USING REAL DB
  CH01LID                   RECON: LIST.DB FOR CS GROUP
  CH01RC1              1    RECOVER MULT. DB'S (CSGRP)
  CH01RC1I             1    RECOVER MULT. DB'S (CSGRP) - IC ONLY
  CH01RCL                   RECOVER MULT. DB'S (LIST)
  CH01RCLI                  RECOVER MULT. DB'S (LIST) - IC ONLY
  CH01RCP                   RECOVER MULT. DB'S (PROMPT)
  CH01RCPI                  RECOVER MULT. DB'S (PROMPT) - IC ONLY
  CH01RC                    RECOVER SINGLE DBD OR DBD/DDN
  CH01RCI                   RECOVER SINGLE DBD OR DBD/DDN - IC ONL
  CH01REI1             1    REORG INDEX DB BY GRP # + IC BY GRP
  CH01REN1             1    REORG INITIALIZE DB BY GRP #
  CH01RE1I             1    REORG PRIME DB + IC BY GRP #
  CH01REBE                  REORG PRIME DB BY DB CODE + IC BY CODE
  CH01RELE                  REORG PRIME DB BY DB CODE + IC BY CODE
  CH01REOE                  REORG PRIME DB BY DB CODE + IC BY CODE
  CH01RE1              1    REORG PRIME DB BY GRP #
  CH01MON                   SPACE MONITOR FOR CS GROUP
****************************** Bottom of data ********************
```

Figure 103. Database Maintenance Tasks Listed by Control Suite

The list acts as a reminder to DBA. When a special task needs to be performed, just select the required task and JCL generates it accordingly. This makes the DBA job easier and is especially helpful for those who are new to OS/390 or IMS DBCTL environment.

The Control Suite has more databases and RECON-related functions, but they are beyond the scope of this book. Refer to the product manual *IMS Database Control Suite User Guide, SC26-9201*, for more information.

10.3 DB Tools

We are introducing another set of tools that can help users improve performance once they become more familiar with IMS DBCTL and OS/390 environment.

IMS System Utilities/DataBase Tools Version 2 (DBT Version 2), Program 5685-093, is a collection of database and online utility products. Customers may or may not need them all (in the DBCTL environment, they are unlikely to need all of them). Therefore, DB Tools is packaged in such a way that several related function utilities are combined into separately orderable features, so customers can pick and chose what fits their own environment.

DB Tools (DBT) is designed to assist users in gaining productivity and performance improvements through better IMS database administration, and through improved IMS database design, installation, maintenance, and tuning. DBT V2 supports both full-function and Fast Path (DEDB) databases. It contains the following features:

- Space management utilities (SMU)
  - HD pointer checker
  - HD tuning aid
  - HDAM physical sequence sort/reload
  - DB segment restructure
  - Space monitor
  - DB historical data analyzer
- High speed sequential retrieval (HSSR)
  - High speed sequential retrieval
  - Sequential subset randomizer
- Fast reorganization reload (FRR)
- Sequential DAM optimizer (SDO)
- DBD/PSB/ACB library management utilities (LMU)
- VSAM zapper
- DEDB pointer checker
- DEDB tuning aid
- DEDB unload/reload
- Fast ACBGEN
- Image copy extensions
- Fast prefix resolution

These tools cover almost every aspect of the IMS system and database management functions. In fact, many of the newer tools products are based on these tools with additional functions to make them more user friendly. It is impossible for us to cover every tool here, so we focus on the simple IMS database back-up and restore functions.

10.3.1 Backup and Restore Sequences

Users who back up their database in *HD format (the default format) with DB Tools find that the backup files are compatible with the standard image copy utilities. Of course the DB Tools dump utility runs faster. However, DB Tools is not a free product. You will have to order and install this product before users can take advantage of it.

10.3.2 Backup of Database

There are two backup DB Tool utilities, FABHURG1 and FABHFSU.

10.3.2.1 FABHURG1

The backup job control language for FABUHRG1 is shown in Figure 104.

```
//BACKUPDB JOB (B3562,W941),'IMS TE-DBDC',
//         MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//        ORDER JCLLIB=IMS.SJIMSD.PROCLIB
//UNLOAD   EXEC FABHULU,MBR=FABHURG1,DBD=dbdname.IMSID=IMSD,DBRC=YES
//SYSPRINT  DD  SYSOUT=*
//SYSUT2    DD  DISP=(NEW,CATLG,DELETE),DSN=IMS.USER.DB.FILE1.BACKUP,
//          UNIT=TAPE,LABEL=(1,SL)
//DFSVSAMP  DD  DISP=DHR,DSN=IMS.SJIMSD.PROCLIB(DFSVSMDB)
//DBD1      DD  DISP=SHR,DSN=USER1.DB.FILE1  Database DD as in DBD.
```

Figure 104. JCL to Run DB Backup Using FABHURG1

See Figure 146 on page 270 for the full job with procedure.

When a HIDAM index is implemented with ISAM/OSAM, the following restrictions apply:

- The index record of the first root must be stored in the ISAM data set.
- The root segment must have twin backward pointers.

  When HISAM is implemented with ISAM/OSAM, it cannot have multiple data set groups.

When unloading a database using a DLI region with a PSB or a DBB region with an ACB, the following restrictions apply.

- The PCB must reference a physical DBD.
- Virtual logical children must not be defined in a PCB SENSEG statement.

A logical child segment with a logical parent′s concatenated key (specified as ″virtual″ on the SEGM statement of the DBD) has blanks returned to the I/O area when the segment is retrieved.

When FABHURG1 is used to unload an uncorrupted database defined as having a logical child with a logical parent′s concatenated key (specified as ″virtual″ on the SEGM statement of the DBD), you must run the IMS Database Prereorganization utility with the control statement DBR= to have a successful reload and prefix resolution. Control statement DBIL= gives incorrect results.

FABHURG1 cannot be used in a reorganization to unload a (broken) database with logical pointer errors that is defined as having a logical child with a logical parent′s concatenated key (specified as *virtual* on the SEGM statement of the DBD).

Since you suspect that logical pointers are invalid, you must run the Database Prereorganization utility using the DBIL= control statement. Therefore, FABHURG1 cannot be used, as you would get an incorrect reload that would be detected during prefix resolution. All related databases should be unloaded and reloaded using the IMS utilities.

FABHURG1 cannot be used in a reorganization to unload a broken database having logical relationships, if either of the following segment occurrences would be skipped or lost during the unload:

- A logical parent segment having one or more logical children.
- A logical child segment that is physically paired.

You must ensure that no segments required by the IMS HD Reorganization. Reload utility are missing during reorganization for either of the following reasons:

- Having any segment types not defined data sensitive in a SENSEG statement. (You are assured this will not happen when unloading using a ULU region.)
- Having a user exit routine that sets a return code indicating so segments should be skipped.

10.3.2.2 FABHFSU

The backup job control language for FABHFSU is shown in Figure 105.

```
//BACKUPDB JOB (B3562,W941),'IMS TE-DBDC',
//         MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//       ORDER JCLLIB=IMS.SJIMSD.PROCLIB
//UNLOAD  EXEC FABHULU,MBR=FABHFSU,DBD=dbdname.IMSID=IMSD,DBRC=YES
//SYSPRINT DD  SYSOUT=*
//SYSUT2   DD  DISP=(NEW,CATLG,DELETE),DSN=IMS.USER.DB.FILE1.BACKUP,
//         UNIT=TAPE,LABEL=(1,SL)
//DFSVSAMP DD  DISP=DHR,DSN=IMS.SJIMSD.PROCLIB(DFSVSMDB)
//DBD1     DD  DISP=SHR,DSN=USER1.DB.FILE1  Database DD as in DBD.
//CARDIN   DD  *
DBDdbdname.

Use to bypass sequence checking but report on errors:
//CARDIN   DD  *
DBDdbdname.       YBY999  1
```

Figure 105. JCL for DB Backup Using FABHFSU

See Figure 146 on page 270 for the full job with procedure.

The following are not supported by FABHFSU:

- Secondary index processing. FABHFSU does, however, support unloading of databases according to the usual HISAM/HIDAM/HDAM root sequences with secondary indexes. The index target segment must be a root.
- Checkpoint or restart. IMS checkpoint or restart can be used by HSSR application programs that use the HSSR database calls.
- Unloading HSAM databases or GSAM databases.
- HISAM ISAM/OSAM databases with multiple data set groups.
- Multivolume data sets that reside on volumes of more than one device type.
- Sensitive, virtually paired segments. This restriction does not preclude unloading of databases for reorganization, since the input data set to IMS HD Reorganization Reload utility does not contain virtually paired segments.

Only limited support is available for HIDAM databases that have a HIDAM index implemented with an ISAM/OSAM. We recommend that a ISAM/OSAM HIDAM index be converted to a KSDS HIDAM index

HIDAM indexes implemented with ISAM/OSAM require that:

- The index record of the first root must be stored in the ISAM data set.
- The root segment must have twin backward pointers.
- The entire database must be unloaded.

For a logical child segment with a logical parent′s concatenated key (specified as ″virtual″ on the SEGM statement of the DBD), binary zeros are returned to the I/O area when the segment is retrieved. (The DBIL control statement is not supported in this case, but the DBR control statement is supported in all cases. See *IMS/ESA, Utilities Reference: Database Manager, SC26-8769*, for a discussion of DBIL versus DBR Reorganizations. For further information on FABHFSU, see manual *IMS System Utilities/Data Base Tools, High Speed Sequential Retrieval Guide, SH21-0548*

10.3.3 Restore

Use the standard Image Copy to restore your database. See Figure 146 on page 270 for an example. Chapter 9, “Operating IMS DBCTL” on page 139 described this process in detail.

Chapter 11. Interoperability

When an application system is large and complex or available resources are limited, migration from one platform to another in one step may be impossible. If so you may want to consider a phased approach of several independent subsystems. You can migrate these subsystems one at a time. Another possible migration method is to migrate the first phase to gain experience, then further divide your resources into groups and carry out the remaining migration phases in parallel.

While there are many ways to divide an application system into phases, all require careful planning, coordination and monitoring. We do not propose to give detailed theoretical software engineering methodology here. Instead, we suggest a few practical procedures. Many of these methods have already been used by what are now production environments. We summarize then here.

Project time constraints prevented us from implementing the entire migration model. However, the model was proved to work correctly at our project design and proof-of-concept times.

11.1 Phased Approach

We assume the application system concerned has the following characteristics:

- It uses a mix of VSAM files and DL/I databases.
- It uses a mix of online transactions and batch programs.
- It has simple transactions and batch programs that use either DL/I databases only or VSAM files only, but not both.
- Any complex transactions and batch programs included may use both DL/I databases and VSAM files.

We can clearly divide this type of application system into the following:

- Simple online transactions that use VSAM files.
- Simple online transactions that use DL/I databases.
- Simple batch programs that use VSAM files.
- Simple batch programs that use DL/I databases.
- Complex online transactions.
- Complex batch programs

Of course, the system can be further divided by other means, such as mapping application languages to available manpower resources. However, we are not going to get into that kind of granularity. Migration, thus, can be carried out by migrating line items one by one, from simple to complex.

11.2 Operating Environment

One problem you face with phased migration is production cut-off when part of the system has already been migrated into OS/390 environment but other parts remain with VSE. In general, we try to have as much operation transparency as possible to avoid forcing the user community (especially for online operation) to learn a new environment.

Our interoperability migration model suggests the following operational staging in addition to the programming conversion mentioned above. We are describing this model in detail in the next few sections.

During the migration processing, the production environment should include both existing VSE operating system as well as OS/390 environment as the integrated production environment until the entire migration is done. Our project environment (Figure 4 on page 7) can be used as an example; it is repeated here.

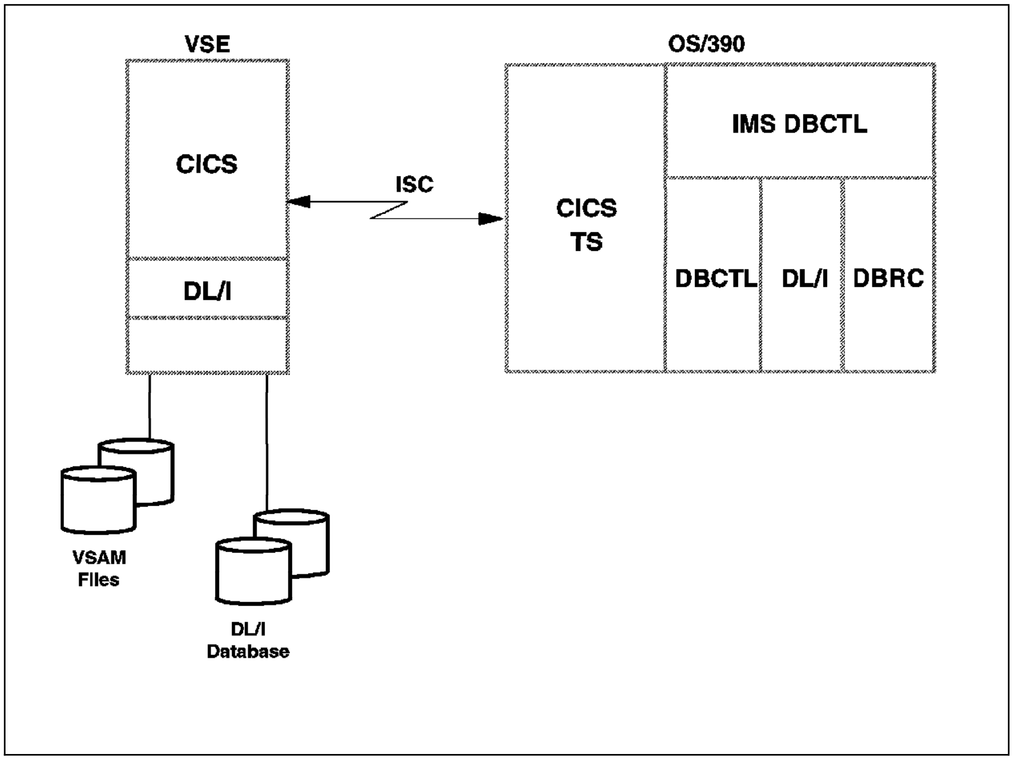


Figure 106. Example of Integrated Production Environment

In the example, we connect the current VSE production CICS region with the newly built OS/390 CICS TS and IMS DBCTL environment. Everything on the VSE side should remain the same except for the addition of the ISC link in the terminal control table (TCT) to connect the OS/390 CICS region. The following illustrates the TCT item:

```
...........

DFHTCT TYPE=SYSTEM,
      ACCMETH=VTAM,
      SYSIDNT=CICD,
      TRMTYPE=LUTYPE62,
      FEATURE=PARALLEL,
      NETNAME=SCSCPAAD,
      CONNECT=AUTO
DFHTCT TYPE=MODESET,
      SYSIDNT=CICD,
      MODENAM=MTAPPC,
      MAXSESS=(10,6),
      CONNECT=AUTO
...........
```

On the OS/390 side, you also need to add an ISC link to connect to the VSE CICS region by adding one group into the start-up parameter GRPLIST. The following is a sample:

```
.........
DEF CONNECTION(CICV) GROUP(OSLU62)
         NETNAME(SAFEII)
         ACCESSMETHOD(VTAM)
         PROTOCOL(APPC)
         AUTOCONNECT(YES)
         INSERVICE(YES)
DEF SESSIONS(SAF1)   GROUP(OSLU62)
         CONNECTION(CICV)
         MODENAME(ISCLU62)
         PROTOCOL(APPC)
         MAXIMUM(10,6)
         SENDSIZE(4096)
         RECEIVESIZE(4096)
         AUTOCONNRCT(YES)
.........
```

No extra work is needed on the IMS DBCTL side. When you start the system, the production environment should be the same as before, except that there are several parallel sessions connecting the empty OS/390 CICS TS and IMS DBCTL application environment.

11.3 Phase One

This phase of migration can be implemented without a need to notify the user community if you use the APPLID of CICS VSE production online region as the newly built OS/390 CICS TS online region APPLID. User log on is from OS/390 CICS TS online region instead of from CICS VSE region. However, all transactions, programs, files, and databases remain in the VSE CICS region and process the work out from CICS VSE. This arrangement lets the operator get to know the basic operation in the OS/390 environment.

To implement this, you need to create a new application transaction resource group. Make all transactions remote transactions and use them to replace the one we migrated from the VSE region. Adjust the CICS start-up parameter GRPLIST accordingly. For example, use the CICS supply transaction CEDA to make the changes. Figure 107 on page 198. shows the Define function panel of the CICS-supplied transaction CEDA.

```
 DEFINE
 ENTER ONE OF THE FOLLOWING
Connection
DB2Conn
DB2Entry
DB2Tran
File
Journalmodel
Lsrpool
Mapset
PARTItionset
PARTNer
PROFile
PROGram
Sessions
TDqueue
TErminal
TRANClass
TRANSaction
TYpeterm


                                              SYSID=CICD APPLID=SCSCPAA

PF 1 HELP      3 END           6 CRSR           9 MSG         12 CNCL
```

Figure 107. DEFINE function of CEDA Transaction Panel

We then set the SAFE II application transaction CZFT as remote to a remote CICS VSE system called CICV (see Figure 108).

```
* Use CEDA DEF Tran:

      TRANSaction  ==> CZFT
      Group        ==> SAFEIIV
      DEscription  ==> SAFE II Online transaction
      PROGram      ==>
      TWasize      ==> 00000             0-32767
      PROFile      ==> DFHCICST
             ................
     REMOTE ATTRIBUTES
      DYnamic      ==> No                No | Yes
      REMOTESystem ==> CICV
      REMOTEName   ==>
      TRProf       ==>
      Localq       ==> No                No | Yes
             ...............
```

Figure 108. Define Remote Transaction Using CDEA

When the user logs on to the CICS TS region CICD and wants to use transaction CZFT, the CICS TS immediately reroutes the work to a remote system, called CICV, to do the processing and send back the result to the user terminal.

11.4 Phase Two and the Continuation

Now, start to move resources from the VSE environment to the OS/390 environment. Following the phased approach mentioned in 11.1, “Phased Approach” on page 195, you can move one or more transactions that use VSAM files or DL/I databases (but not both) over.

When a transaction is moved over, activate it by adjusting the transaction definition as follows:

- Add the name of the application program used by the transaction.

```
 TRANSaction  ==> CZFT
 Group        ==> SAFEIIV
 DEscription  ==> SAFE II Online transaction
 PROGram      ==> CZFHSM
        ................
```

- Reset the REMOTE attributes.

```
        ...............
REMOTE ATTRIBUTES
 DYnamic      ==> No                   No | Yes
  REMOTESystem ==>
 REMOTEName   ==>
        ...............
```

  Restart CICS TS to use the new attributes.

- If the transaction uses DL/I databases and databases have already been reloaded to OS/390, it should start automatically. If not, issue IMS command:

```
/STA DB <database name>
```

  We assume here that related database and application macros have already been coded in the IMS DBCTL region Stage 1 source. MODBLKS generation, on-line change and database dynamic allocation modules, if any, have already been performed as described in 8.6.7, “Change CICS DASD Logs to MVS Logger DASD Log Streams” on page 101.

- If the transaction uses VSAM files and files have been exported to OS/390, adjust the SIT start-up parameter to reflect the new environment. Use RDO to install the FCT group if necessary.

At a certain point, the production environment may look like Figure 109 on page 200.

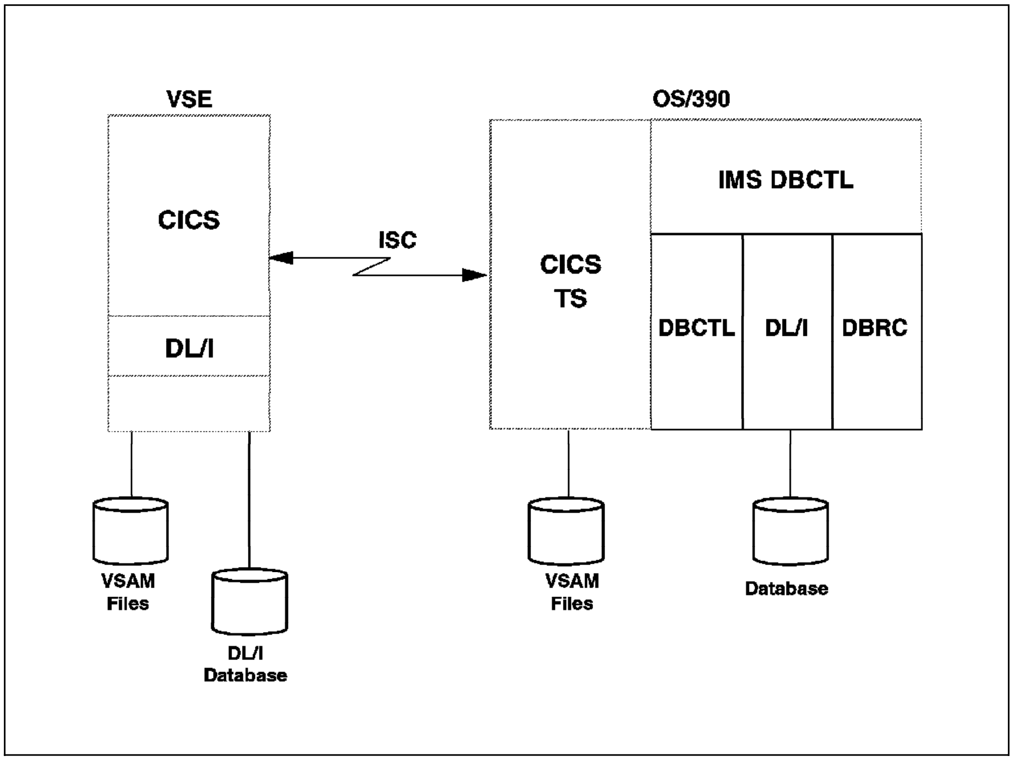


Figure 109. Example of Phase Two Integrated Production Environment

The figure shows that some resources could already be moved to OS/390 and some still remain on the VSE side. However, the user community need not know where the processes are actually taking place.

11.5 Final Phase

When all work has been moved to OS/390, the ISC link between the two environments can be dropped by deleting the linkage group from the CICS TS parameter GRPLIST. You can even do it online by setting the link to *OUTservice*

11.6 Conclusion

Migration by phased approach has its advantages as well as disadvantages:

- Advantages
  - It will be easy to locate application problems since the phased migration narrows down the debugging scale.
  - Phased migration allows DBA monitoring and system tuning immediately. Also, phasing makes it much easier to tune the system.
  - Phased migration uses much less resources, is easier to manage, and facilitates recovery.
  - Members of the migration team can concentrate on one technical area at a time, making it much easier to acquire the necessary new skill.
- Disadvantages
  - Phased migration takes extra time to get the work done, prolonging the migration time scale.

– Extra management and coordination are needed.
– Resource can be tied up for longer periods.

Chapter 12. Problem Determination

The first difficulty you face in resolving a problem is trying to determine where to look for it. In the case of CICS DL/I, you would have switched on the CICS trace. The trace data is written to the auxiliary trace data sets, and then formatted. That way, the application′s calls to CICS and DL/I, along with many of the CICS internal functions, are all available in one place. With CICS TS and IMS/ESA DBCTL, much more information is available, but you also have a harder job to decide which pieces of information are of use.

12.1 DBCTL Tracing

The following describes how the IMS DBCTL trace function works. For details, there is IMS manual *IMS/ESA, Operations Guide, SC26-8741*.

12.1.1 The IMS Trace Command

The IMS trace command is issued by the command TRACE. The format from CICS CDBM is shown in Figure 110.

```
                 |-ON--|
_/TRACE_ __SET__|_____|_____________________________________________
_/TRA___|        |_OFF_|

  >__
     |                                                                  |
     |_MONITOR____ALL___________________________________________________|
     |_OSAMGTF__________________________________________________________|
     |_PI__ _________________ __ ____________________ _______________|
     |      |        _LOG__  |  |           _MEDIUM_  |                 |
     |      |_OPTION__|______|_|  |_VOLUME__|_________|_|                 |
     |                |_TIME_|              |_LOW____|                    |
     |                |_ALL__|              |_HIGH___|                    |
     |             <_________________                                    |
     |_PROGRAM____programname_|_________________________________________|
     |                _NOCOMP_                                           |
     |_PSB__psbname__|________|_________________________________________|
     |                |_COMP___|                                         |
     |                                                                  |
  >__|_TABLE__ _ALL__ __ _________________ __ __________________ _
              |_DISP_|  |          _NOLOG_  |  |          _MEDIUM_  |
              |_DL/I_|  |_OPTION__|_______|_|  |_VOLUME__|_________|_|
              |_DLOG_|            |_LOG___|              |_LOW____|
              |_FAST_|                                   |_HIGH___|
              |_LATC_|
              |_LOCK_|
              |_LRTT_|
              |_RETR_|
              |_SCHD_|
              |_STRG_|
              |_SUBS_|
```

Figure 110. IMS TRACE Command Syntax for DBCTL

You can issue the command either through the CICS CDBM interface, or through the operating console using the CRC or IMSID as a prefix to the command instead of the '/'. The IMS manual offers many other options, but the ones in Figure 110 are supported for DBDC. For example, use these to set the OSAM GTF trace on:

```
/TRA SET ON OSAMGTF        from CICS CDBM
>TRA SET ON OSAMGTF        from SYSLOG master console with crc
/>TRA SET ON OSAMGTF       from SDSF using crc
IMSDTRA SET ON OSAMGTF     from SYSLOG master console with IMSid
/IMSDTRA SET ON OSAMGTF    from SDSF using IMSid
```

To set the OSAM GTF trace off, even if no other traces are active, use these:

```
/TRA SET OFF OSAMGTF       from CICS CDBM
>TRA SET OFF OSAMGTF       from SYSLOG master console with crc
/>TRA SET OFF OSAMGTF      from SDSF using crc
IMSDTRA SET OFF OSAMGTF    from SYSLOG master console with IMSid
/IMSDTRA SET OFF OSAMGTF   from SDSF using IMSid
```

12.1.2 IMS Traces Available

There are several different traces you can use.

12.1.2.1 IMS Monitor

This is covered in 12.4, “Program Isolation” on page 213

12.1.2.2 OSAMGTF Trace

You can set the OSAMGTF trace on, by typing the command

```
/TRA SET ON OSAMGTF.
```

This starts GTF tracing for OSAM buffers, if you defined it in the member IMS.SJIMSD.PROCLIB(DFSVSMxx), where xx =IB in our sample setup. The line is

```
OSAMOP  OSAMGTF=YES
```

If no tracing is defined, then even switching this trace on has no effect, and no GTF trace is generated.

12.1.2.3 Program Isolation Trace

The program isolation (PI) trace can be started with several options. It records information about program isolation ENQ/DEQ calls and DL/I calls. These records are written to the same trace table as DL/I and LOCK entries. Starting the LOCK table trace automatically starts the PI trace. It records ENQ/DEQ events that occur when DL/I calls are made. It can be started with several options, as shown in Table 17 on page 205.

Table 17. PI Trace Options	
PI Trace command	**Effect of option**
/TRACE SET ON PI.	PI records written to in-store trace table.
/TRACE SET ON PI OPTION LOG.	PI records written to OLDS data set record type X′67FA′.
/TRACE SET ON PI OPTION TIME.	PI records written with extra field for elapsed time if the ENQ/DEQ had to wait.
/TRACE SET ON PI OPTION .... VOLUME HIGH/MEDIUM/LOW	Number of records written.
/TRACE SET ON PI OPTION ALL. VOLUME HIGH/MEDIUM/LOW	Effect of OPTION LOG and OPTION TIME.
/TRACE SET OFF PI.	Switches all PI tracing off.

12.1.2.4 Program Trace

This is mainly used to trace IMS DC calls, so provides very little information in a DBCTL environment.

12.1.2.5 PSB Trace

The PSB trace traces all DL/I calls made against each PSB. It writes out x′5F′ type records to the IMS OLDS data set.

12.1.2.6 Trace Tables

This writes out a number of various traces to the trace table. The OPTION LOG and VOLUME MEDIUM/HIGH/LOW have the same meanings as for PI trace. The meanings of the various trace tables are shown in Table 18.

Table 18. TRACE Tables	
/TRA TABLE type	**What is traced**
/TRACE SET ON TABLE DISP	Dispatcher trace.
/TRACE SET ON TABLE DL/I.	DL/I call trace
/TRACE SET ON TABLE DLOG.	Logging trace.
/TRACE SET ON TABLE FAST.	Fast Path Trace.
/TRACE SET ON TABLE LATC.	Latch trace.
/TRACE SET ON TABLE LOCK.	Lock trace.
/TRACE SET ON TABLE LRTT.	Log Router Trace in RSR.
/TRACE SET ON TABLE RETR.	DL/I Retrieval trace.
/TRACE SET ON TABLE SCHD.	Scheduler trace.
/TRACE SET ON TABLE STRG.	Storage subsystem trace
/TRACE SET ON TABLE SUBS.	External subsystem trace.
/TRACE SET ON TABLE ALL.	All the above.
/TRACE SET OFF TABLE ALL.	Switch all trace tables off.

A good set of traces, able to meet most of your needs, is:

```
/TRA SET ON TABLE DISP.
/TRA SET ON TABLE DL/I.
/TRA SET ON TABLE LOCK.
/TRA SET ON TABLE STRG.
/TRA SET ON TABLE SCHD.
```

12.1.3 How to Collect IMS Trace Data

IMS trace data is written out as log records of several types, including x′67′ and x′5F′ types. The data is written out to the trace data set or to the current OLDS if trace data sets are not available. IMS preferences are in the following order:

1. DFSTRA01 or DFSTRA02 data sets on disk, if catalogued and allocated, and if a DD statement or dynamic allocation member is present.
2. DFSTRA0T data set on tape — not catalogued — if a DD statement or dynamic allocation member is present.
3. Current OLPnn and OLSnn data set.

12.1.4 How to Print Out an IMS Trace

Run a job like that in Figure 111 to print out all the trace records.

```
* JCL found in member IMS.SJIMSD.JOBS(TRAPRINT)

//     SET SYS=SJIMSD
//*
//* JOB FOR IMSGEN CHECKLIST - PROCESS LOG RECORDS
//*
//DFSERA10 EXEC PGM=DFSERA10
//STEPLIB   DD  DISP=SHR,DSN=IMS.&SYS..RESLIB
//SYSPRINT  DD  SYSOUT=*
//SYSUT1    DD  DISP=SHR,DSN=IMS.&SYS..DFSTRAOT,BUFNO=40
//SYSIN     DD  *
*
* DUMP RECORDS
*
CONTROL  CNTL
OPTION   PRINT OFFSET=5,FLDLEN=1,VALUE=67,COND=E,EXITR=DFSERA30
OPTION   PRINT OFFSET=5,FLDLEN=1,VALUE=5F,COND=E,EXITR=DFSERA30
//
```

Figure 111. Sample JCL to Print IMS Trace from DFSTRA0x

This JCL produces a print-out of the trace records in the data set. You then have to interpret them. See *IMS/ESA, Diagnosis Guide and Reference, LY37-3731*

12.2 How to Activate CICS TS Trace

The CICS TS trace is activated either at start-up or by a CICS CEMT command. It may also be initiated by some third-party products.

12.2.1 Start-up Options

The CICS SIT parameters options are shown in Table 19.

Table 19. CICS Trace — SIT Parameters Options

SIT start-up option	Effect
AUXTR=ON/OFF	Whether trace data is written (ON) or not (OFF) to auxiliary trace data sets.
AUXTRSW=NO/ALL/NEXT	Whether auxiliary trace data sets are switched whenever one fills (ALL) or once only (NEXT) or not at all (NO)
GTFTR=ON/OFF	Whether CICS trace data is written to MVS GTF (ON) or not (OFF).
INTTR=ON/OFF	Whether CICS is to write to its trace table (ON) or not (OFF).
SPCTR=1/(1,2)/(1,2,3)/ALL/OFF	The level of special tracing for all CICS components.
SPCTRxx=1/(1,2)/(1,2,3)/ALL/OFF	The level of special tracing for CICS xx component. (See 12.1.1, "The IMS Trace Command" on page 203)
STNTR=1/(1,2)/(1,2,3)/ALL/OFF	The level of standard tracing for all CICS components.
STNTRxx=1/(1,2)/(1,2,3)/ALL/OFF	The level of standard tracing for CICS xx component. (See 12.1.2, "IMS Traces Available" on page 204)
SYSTR=ON/OFF	Specifies whether standard tracing is written (ON) or not (OFF)
TRTABSZ=nnnn	Size of CICS Trace table in kilobytes.

12.2.2 CICS Commands

The following are the possible CICS TS trace commands:

- CEMT I/S to inquire about or set the trace on:

  AUXTR Auxiliary

  GTFTRACE Starts and stops GTF trace

  INTTRACE Starts and stops Internal trace (compare CEMT S TRACE on VSE)

- CETR to set all the trace options (see Figure 112 on page 208).

```
CETR               CICS Trace Control Facility                 CICD

Type in your choices.

Item                             Choice       Possible choices

Internal Trace Status       ===>  STARTED      STArted, STOpped
Internal Trace Table Size   ===>  16      K    16K - 1048576K

Auxiliary Trace Status      ===>  STOPPED      STArted, STOpped, Paused
Auxiliary Trace Dataset     ===>  A            A, B
Auxiliary Switch Status     ===>  NEXT         NO, NExt, All

GTF Trace Status            ===>  STOPPED      STArted, STOpped

Master System Trace Flag    ===>  ON           ON, OFf
Master User Trace Flag      ===>  ON           ON, OFf

When finished, press ENTER.




PF1=Help      3=Quit      4=Components      5=Ter/Trn     9=Error List
```

Figure 112. CETR Screen — Main Screen

When you have selected the options you want on the main screen, the screen shown in Figure 113 appears.

```
 CETR                    Component Trace Options                 CICD

 Over-type where required and press ENTER.                       PAGE
 Component Standard                          Special
 --------  -------------------------------   --------------------------
   AP      1                                 1-2
   BM      1                                 1
   BR      1                                 1-2
   CP      1                                 1-2
   DC      1                                 1
   DD      1                                 1-2
   DM      1                                 1-2
   DS      1                                 1-2
   DU      1                                 1-2
   EI      1                                 1-2
   FC      1                                 1-2
   GC      1                                 1-2
   IC      1                                 1-2
   IS      1                                 1-2
   KC      1                                 1
   KE      1                                 1-2
   LC      1                                 1-2

PF1=Help      3=Quit      4=Components      5=Ter/Trn     9=Error List
```

Figure 113. CETR Screen — Trace Levels for Components

The final screen, on which you select the individual terminal or transaction trace to be obtained is shown in Figure 114 on page 209.

```
CETR                    Transaction and Terminal Trace           CICD

Type in your choices.

Item                              Choice       Possible choices

Transaction ID              ===>               Any valid 4 character ID
Transaction Status          ===>               STandard, SPecial, SUppr

Terminal ID                 ===>               Any valid Terminal ID
Netname                     ===>               Any valid Netname
Terminal Status             ===>               STandard, SPecial
Terminal VTAM Exit Trace    ===>               ON, OFf
Terminal ZCP Trace          ===>               ON, OFf

VTAM Exit override          ===>  NONE         All, System, None


When finished, press ENTER.




PF1=Help      3=Quit      6=Cancel Exits      9=Error List
```

Figure 114. CETR Screen — Individual Terminal or Transaction Trace

12.2.3 Types of CICS TS Trace

There are three types of CICS tracing available:

- **System tracing**. This is always switched on even if internal trace switched off.
- **Standard tracing**. This is tracing that is done throughout the lifetime of a transaction.
- **Special tracing**. This provides much more data, but only in special circumstances. These traces are made at CICS exception points, that is, at program abend. This includes a dump of the registers and certain areas of storage.

There are three levels of tracing.

- **Level 1**. On entry and exit from each:
  - CICS domain.
  - Major domain function.
  - External functions (such as VTAM).
  - Location where there was a trace point in CICS Version 2.
- **Level 2**. Points between the Level 1 trace points. This is used for debugging user programs. If Level 2 is specified, Level 1 must also be specified.
- **Level 3**. Normally used only under the direction of IBM support staff.

All of the components listed in Table 20 on page 210 can be traced. Supply their two-letter code (xx) to the start-up options:

Table 20. Domain Names Can Be Traced by Command CICS TRACE			
Code	**Command Name**	**Code**	**Command Name**
AP	Application domain	BM	Basic mapping support
BF	Built-in function	BR	3270 Bridge
CP	Common programming interface	DC	Dump compatibility layer
DD	Directory manager domain	DI	Batch data interchange
DM	Domain manager domain	DS	Dispatcher domain
DU	Dump domain	EI	Exec interface
EX	External CICS interface	FC	File control
GC	Global catalog domain	IC	Interval control
IS	Intersystem communication	KC	Task control
KE	Kernel	LC	Local catalog domain
LD	Loader domain	LG	Log manager domain
LM	Lock domain	ME	Message domain
MN	Monitoring domain	NQ	Enqueue domain
PA	Parameter domain	PC	Program control
PG	Program manager domain	RI	Resource manager interface (RMI)
RM	Recovery manager domain	SC	Storage control
SM	Storage domain	ST	Statistics domain
SZ	Front-end programming interface	TC	Terminal control
TD	Transient data	TI	Timer domain
TR	Trace domain	TS	Temporary storage domain
UE	User exit interface	US	User domain
WB	WEB interface	XM	Transaction manager domain
XS	Security manager domain		

12.2.4 How to Collect CICS Trace

CICS trace data is written out to the auxiliary trace data sets. These are two tape or disk data sets with the DD names DFHAUXT and DFHBUXT. As in CICS TS, they should be allocated using JCL if you intend to use them. They should typically have a DCB attribute of DCB=(BLZSIZE=4096,LRECL=4096,RECFM=F). You also get a dump of the trace table in a CICS dump, such as

```
//DFHAUXT  DD  DSN=CICS.SJCICSD.DFHAUXT,DISP=SHR,DCB=BUFNO=5
//DFHBUXT  DD  DSN=CICS.SJCICSD.DFHBUXT,DISP=SHR,DCB=BUFNO=5
```

12.2.5 How to Print the CICS Trace

To print out the trace from the auxiliary trace data set, run a job similar to that in Figure 115.

```
*  JCL found in member IMS.SJIMSD.JOBS(TRAPRINT)

//         SET SYS=SJCICSD
//*
//* JOB FOR IMSGEN CHECKLIST - PROCESS LOG RECORDS
//*
//TRACEPRT EXEC PGM=DFHTU520
//STEPLIB   DD  DISP=SHR,DSN=CICS.&SYS..SDFHLOAD
//SYSPRINT  DD  SYSOUT=*
//DFHAXPRT  DD  SYSOUT=*
//DFHAUXT   DD  DSN=CICS.&SYS..DFHAUXT,DISP=SHR
//*DFHAUXT  DD  DSN=CICS.&SYS..DFHBUXT,DISP=SHR
//*   Use the default parameters for a full print of the auxiliary
//*   trace dataset. Selection parameters go in DFHAXPRM.
//DFHAXPRM  DD  *
/*
//
```

Figure 115. Sample JCL to Print CICS Trace from Auxiliary Trace Data Sets

If you need to select only some of the records, code selection commands as described in M.4, "How to Specify Trace Data to Print" on page 326 or see the manual *CICS Transaction Server for OS/390 CICS Operation and Utility Guide, SC33-1685* for details of the parameters.

12.3 IMS Log Records

IMS writes log records for many different events, including all events on any database defined to IMS. These records can be hard to interpret, and the format differs from record to record. However only a few are applicable in a DBCTL environment.

12.3.1 IMS Log Record Types

The principal log record types users want to look at for diagnostic purposes are shown in Table 21 on page 212.

Table 21. Principal IMS Log Record for Diagnostics	
Record type.	**Purpose.**
x'06'	IMS was started or stopped.
x'07'	A dependent region (CICS thread or BMP) was started or stopped.
x'08'	An application program was scheduled.
x'10'	A security violation occurred.
x'20'	A database was opened.
x'21'	A database was closed.
x'24'	An I/O error was detected by the buffer handler.
x'37'	Sync point reached. Phase 1 of two-phase commit completed.
x'37'	Sync point aborted and backed out. Phase 2 of two-phase commit failed.
x'41'	BMP checkpoint record.
x'4C'	Database was stopped or started, or an I/O error occurred, or a program was stopped.
x'50'	A database was updated. Record contains the after image. It also contains the old data for a delete call.
x'51'	A database record was replaced. Record contains the record before it was replaced.
x'52'	An insert was done for a new root segment in a key sequence database. This contains the key sequence data set record before change.
x'5F'	Trace record written when a DL/1 call completes.
x'67'	General record. Many are trace records. Very useful for diagnostics.
x'6701'	/TRACE command was issued.
x'6705'	Dependent region (BMP or CICS thread) abnormally terminated.
x'67FA'	Trace record.
x'67FB'	Record for Master Console that could not be sent. (Console is not available)
x'67FF'	Dependent region (BMP or CICS thread) abnormally terminated.
x'70'	/Modify command was completed successfully. MODSTAT data set updated.

12.3.2 How to Print IMS Log Records

If you have a problem to investigate, first ensure that the data has all been archived from the OLDS to an SLDS. Do this by issuing the command

```
/SWI OLDS.
```

to IMS DBCTL. Wait until the archive job ARCHIMSD has completed. Then run a job similar to that in Figure 116 on page 213

```
*    JCL found in member IMS.SJIMSD.JOBS(LOGPRINT)

//      SET SYS=SJIMSD
//*
//* JOB FOR IMSGEN CHECKLIST - PROCESS LOG RECORDS
//*
//DFSERA10 EXEC PGM=DFSERA10
//STEPLIB   DD  DISP=SHR,DSN=IMS.&SYS..RESLIB
//SYSPRINT  DD  SYSOUT=*
//SYSUT1    DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*   )  Make sure
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*   )  these are
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*   )  in ascending
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*   )  date/time
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*   )  order, oldest
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*      at top of
//          DD  DISP=SHR,DSN=IMS.&SYS..SLDSP.D*.T*.V*      list.
//          DD  .........................................
//SYSIN     DD  *
*
* DUMP RECORDS
*
CONTROL  CNTL
OPTION   PRINT OFFSET=5,FLDLEN=1,VALUE=07,COND=E,EXITR=DFSERA30
OPTION   PRINT OFFSET=5,FLDLEN=2,VALUE=6705,COND=E,EXITR=DFSERA30
OPTION   PRINT OFFSET=5,FLDLEN=2,VALUE=67FF,COND=E,EXITR=DFSERA30
```

Figure 116. Sample JCL to Print IMS Trace from IMS SLDSP Logs

12.4 Program Isolation

DBCTL forces implicit program isolation by running each program in its own region. Therefore, data from one IMS region, MPP or BMP, can interfere with data from another only if it is written to the MVS CSA. CICS TS achieves a high degree of program isolation by the use of storage protection keys.

12.4.1 CICS TS Program Isolation

CICS/TS has five dynamic storage area (DSAs) compared with CICS/VSE′s one. These are:

- SDSA, where only CICS modules run
- RDSA, read-only DSA. You can place reentrant programs here to run in 24-bit mode. This storage cannot be updated.
- ERDSA, read-only DSA. You can place reentrant programs here to run in 31-bit mode. This storage cannot be updated.
- UDSA, like CICS/MVS V2 DSA. Anything else goes here, in 24-bit mode. This is an application program′s usable storage area; however, CICS can still protect it by 256 storage keys.
- EUDSA, like CICS/MVS V2 DSA. Anything else goes here, in 31-bit mode. Again, an application program′s usable area, but CICS protects it by 256 storage keys.

You can affect where a program goes by some parameters. These include setting the following SIT parameters, in CICS.SJCICSD.PARMLIB(OVERRIDE) in our case:

- CHKSTSK={ALL|CURRENT|NONE}. Defines whether a task-storage violation check is on or not.

  ALL All storage areas on the transaction storage chains are to be checked for all tasks. are to be checked for all tasks.

  CURRENT All storage areas on the transaction storage chain are to be checked for the current task only.

  NONE Task storage-violation checking is to be deactivated. (You can also use the CICS-supplied transaction, CSFE, to switch task storage violation checking on and off.)

- CWAKEY=USER|CICS. Determines whether the CWA uses CICS or USER storage key.
- STGPROT=NO|YES. Determines whether storage protection is active or not.
- TCTUAKEY=USER|CICS. Determines whether TCTUAs are USER or CICS storage key.
- TCTUALOC=BELOW|ANY. Determines whether TCTUAs are below the line or not.
- TRANISO=NO|YES.

  Yes Means that CICS can use over 256 storage keys to isolate transactions from each other if STGPROT=YES.

  No Means that only USER and CICS keys are separated.

By using these keys, you can separate CICS internal transactions from user transactions, and also separate user transactions from each other. It also needs the requisite hardware to enable this. However, we advise running STGPROT=NO, initially.

The isolation can also be achieved by setting the parameter EXECKEY=USER|CICS in the program definition. It indicates whether the program should run in CICS or USER DSA.

EXECKEY=CICS. Programs can access any storage.

EXECKEY=USER. Can only access user storage.

Another place for setting parameters is in the transaction definition:

- TASKDATAKEY=CICS|USER. Determines whether task data is CICS or USER key.
- TASKDATALOC=BELOW|ANY. Determines whether task data is below the line or not
- ISOLATE=YES|NO.
  - If ISOLATE=YES, then this task has its own storage key,
  - If STGPROT=YES, no other task running in USER storage can access its storage.

– If ISOLATE=NO, then any task can share this task's storage.

Table 22 tries to illustrate the relationship of program isolation with various SIT parameters, program definition parameter and transaction definition parameters.

Table 22. Cross Reference of Which Program's Storage Can be Overwritten

STGPROT	**Overwriting task**	**Overwritten task**	**Storage overwrite allowed?**
STGPROT=YES	EXECKEY=CICS	EXECKEY=CICS	YES
STGPROT=YES	EXECKEY=CICS	EXECKEY=USER ISOLATE=YES	YES
STGPROT=YES	EXECKEY=CICS	EXECKEY=USER ISOLATE=NO	YES
STGPROT=YES	EXECKEY=USER ISOLATE=YES	EXECKEY=USER ISOLATE=YES	NO
STGPROT=YES	EXECKEY=USER ISOLATE=YES	EXECKEY=USER ISOLATE=NO	YES
STGPROT=YES	EXECKEY=USER ISOLATE=NO	EXECKEY=USER ISOLATE=YES	NO
STGPROT=YES	EXECKEY=USER ISOLATE=NO	EXECKEY=USER ISOLATE=NO	YES
STGPROT=NO	EXECKEY=CICS	EXECKEY=CICS	YES
STGPROT=NO	EXECKEY=CICS	EXECKEY=USER ISOLATE=YES	YES
STGPROT=NO	EXECKEY=CICS	EXECKEY=USER ISOLATE=NO	YES
STGPROT=NO	EXECKEY=USER ISOLATE=YES	EXECKEY=USER ISOLATE=YES	YES
STGPROT=NO	EXECKEY=USER ISOLATE=YES	EXECKEY=USER ISOLATE=NO	YES
STGPROT=NO	EXECKEY=USER ISOLATE=NO	EXECKEY=USER ISOLATE=YES	YES
STGPROT=NO	EXECKEY=USER ISOLATE=NO	EXECKEY=USER ISOLATE=NO	YES

12.4.2 CICS Transaction Abends

CICS abend codes in CICS/TS are the same as those in CICS/VSE. They are stored in the EIB, although the location of the EIB in a dump is slightly different. Refer to the product publications for more information.

12.5 COBOL Debugging

There are several ways to debug an application transaction for CICS. They are summarized here.

12.5.1 IMS Causes a CICS Transaction Abend

If IMS causes CICS to abend a transaction, then if the CICS thread is purged in IMS, CICS writes an SVC dump to the MVS dump data sets and continues, without the failed thread.

If the thread simply hits a DL/I return code, indicating a DL/I failure, the CICS transaction will abend and dump.

12.5.2 Print an Abending Program Dump

You can print the CICS dump data set by issuing the command

```
CEMT S DUMPDS SWI
```

Then run the job like Figure 118 on page 217.

```
//      SET SYS=SJCICSD
//*
//* JOB FOR IMSGEN CHECKLIST - PROCESS LOG RECORDS
//*
//DUMPPRNT EXEC PGM=DFHDU520
//STEPLIB   DD  DISP=SHR,DSN=CICS.&SYS..SDFHLOAD
//SYSPRINT  DD  SYSOUT=*
//DFHPRINT  DD  SYSOUT=*
//DFHTINDX  DD  SYSOUT=*
//DFHDMPDS  DD  DSN=CICS.&SYS..DFHDMPA,DISP=SHR
//*DFHDMPDS DD  DSN=CICS.&SYS..DFHDMPB,DISP=SHR
//*   USE THE DEFAULT PARAMETERS FOR A FULL PRINT OF THE AUXILIARY
//*   TRACE DATASET. SELECTION PARAMETERS GO IN DFHAXPRM.
//SYSIN DD  *
/*
//
```

Figure 117. Sample JCL to Print CICS Dump Data Sets

12.5.3 Print CICS System Abend Dump

System dumps in CICS are no longer written to the CICS dump data sets in CICS TS. They are now written to system dump data sets. In OS/390 V2.4, MVS allocates a new data set for each dump. To find its name, look at the MVS console SYSLOG for messages of the form shown:

```
IEA794I SVC DUMP HAS CAPTURED: 982
DUMPID=018 REQUESTED BY JOB (CICSD   )
DUMP TITLE=CICS DUMP: SYSTEM=SCSCPAAD CODE=MT0001   ID=9/0001

IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00048)
IEF196I         DSN (DUMP.D0508.H00.SC62.CICSD.S00018           )
IEF196I         STORCLAS (SCDUMP) MGMTCLAS (STANDARD) DATACLAS (
IEF196I )
IEF196I         VOL SER NOS= DUMPS1
F CICSD,CEMT PERF SHUT IMM
IEF196I IGD104I DUMP.D0508.H00.SC62.CICSD.S00018             RETAINED,
IEF196I DDNAME=SYS00048
IEA611I COMPLETE DUMP ON DUMP.D0508.H00.SC62.CICSD.S00018 993
DUMPID=018 REQUESTED BY JOB (CICSD   )
FOR ASID (0079)
INCIDENT TOKEN: WTSCPLX1 SC62     05/08/1998 00:30:34
```

where SCSCPAAD is the CICS APPLID, and DUMP.D0508.H00.SC62.CICSD.S00018 is the name of the data set containing the dump. You must use IPCS to read this dump. There are two ways to use IPCS to print a CICS system dump.

12.5.3.1 IPCS from ISPF

You can use TSO, or, as discussed next, you can use batch mode. First, log onto TSO, then find the IPCS option in ISPF. This gives you a screen like that in Figure 118.

```
------------------------ IPCS PRIMARY OPTION MENU  --------------------
OPTION  ===>
                                                           ************
   0  DEFAULTS    - Specify default dump and options       * USERID  -
   1  BROWSE      - Browse dump data set                   * DATE    -
   2  ANALYSIS    - Analyze dump contents                  * JULIAN  -
   3  UTILITY     - Perform utility functions              * TIME    -
   4  INVENTORY   - Inventory of problem data              * PREFIX  -
   5  SUBMIT      - Submit problem analysis job to batch   * TERMINAL-
   6  COMMAND     - Enter subcommand, CLIST or REXX exec   * PF KEYS -
   T  TUTORIAL    - Learn how to use the IPCS dialog       ************
   X  EXIT        - Terminate using log and list defaults

Enter END command to terminate IPCS dialog



 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 118. IPCS Initial Menu

Select Option 5 to submit a job. This brings up the screen shown in Figure 119 on page 218.

```
------------------- IPCS MVS DUMP BATCH JOB OPTION MENU    -------------
OPTION  ===>
                                                           *************
   1  SADUMP     - Prepare stand alone dump for analysis   * USERID  - I
   2  SVCDUMP    - Prepare SVC dump for analysis           * DATE    - 9
   3  SYSMDUMP   - Prepare SYSMDUMP for analysis           * JULIAN  - 9
   4  SUPPLEMENT - Perform supplementary dump analysis     * TIME    - 2
   5  EREP       - Process software data using EREP        * PREFIX  - I
                                                           * TERMINAL- 3
JOB STATEMENT INFORMATION:  (Verify before proceeding)     * PF KEYS - 2
                                                           *************
  ===> //*
  ===>
  ===>
  ===>
  ===>
  ===>

Enter END to terminate batch job processing.

  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
  F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 119. IPCS Second Initial Menu

On this second menu, select Option 4 for supplementary data. This brings up the screen in Figure 120.

```
 ------------------ Perform Supplementary IPCS Dump Analysis -----------
 COMMAND ===>

 Enter/verify parameters for the job.
 Use ENTER to submit the job, END to terminate without job submission.

 DATA SET NAME  ===> 'DUMP.D0508.H00.SC62.CICSD.S00018'
 DUMP DIRECTORY ===> DDIR
 SYSOUT CLASS   ===> A

 IPCS SUBCOMMAND, CLIST or REXX exec:
 ===> VERBEXIT CICS520 'CSA=2,DEF=1,DLI=3,TRS=<TRANID=CECI,KE_NUM=12>'

 ADDITIONAL CLIST or REXX exec LIBRARIES: (optional)
 ===>
 ===>




  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=RFIND   F6=MORE    F7=UP
  F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 120. IPCS Job Submission Screen

You need to enter the dump data set name, found above from the MVS SYSLOG output, and the IPCS commands to format the dump. In the example in Figure 120, we want to print out all control blocks in summary, print out the DLI blocks in full and with a summary, and print out the CSA in dump format only. In addition, we want trace entries for all CECI transactions, and for tasks with a kernel number x'0012'. Pressing Enter submits the job.

12.5.3.2 IPCS in Batch

You can, as alternative, submit a job like that in Figure 121 to print out the dump.

```
*  JCL in member CICS.SJCICSD.DBDC.JCL(IPCSPRNT)

//IMSVSE1I  JOB ACCNT#,
//             IMSVSE1,       **JOB STATEMENT GENERATED BY SUBMIT**
//             NOTIFY=IMSVSE1,
//             MSGLEVEL=(1,1)
//*
//*-------------------------------------------------------------------
//* 1. PERFORM IPCS DUMP INITIALIZATION AGAINST A SYSMDUMP.
//* 2. PRODUCE A STANDARD PROBLEM SCREENING REPORT REGARDING THE DUMP
//*-------------------------------------------------------------------
//BLSKBSYS EXEC PGM=IKJEFT01,REGION=5000K
//SYSPROC   DD  DSN=SYS1.SBLSCLI0,DISP=SHR
//SYSTSPRT  DD  SYSOUT=*,
//          DCB=(RECFM=VB,LRECL=124,BLKSIZE=4096)
//IPCSDDIR  DD  DSN=IMSVSE1.DDIR,DISP=SHR
//IPCSDUMP  DD  DSN=DUMP.D0508.H00.SC62.CICSD.S00018,DISP=SHR
//IPCSTOC   DD  SYSOUT=*
//IPCSPRNT  DD  SYSOUT=*
//SYSTSIN   DD  *
%BLSCBSUQ DUMP.D0508.H00.SC62.CICSD.S00018
IPCS NOPARM
EVALDEF CLIST(CONFIRM(CON) PRINT(PRT) SOURCE(SRC) TERMINAL(TRM)
    FLAG(FLG))    LOCAL
SETDEF DSNAME('DUMP.D0508.H00.SC62.CICSD.S00018') NOCONFIRM PRINT -
        NOTERMINAL FLAG(ERROR) LOCAL
VERBEXIT CICS520 'CSA=2,DEF=3,DLI=1,TRS=<TRANID=CSSC,KE_NUM=12>'
/*
//
```

Figure 121. IPCS Job Submission Screen

If you use this parameter:

```
VERBEXIT CICS520 'DEF=3'
```

the system prints out all the control blocks and their summary, which will do to get started with. You can become more selective later. The IPCS commands for CICS are given in Appendix M, "IPCS Commands for Formatting CICS Dumps" on page 323.

You can obtain unformatted dumps in CCTLSNAP if you include:

```
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

statements in the CICS region JCL. However, you cannot then use IPCS.

12.5.4 IPCS Online

You can also use the IPCS function of ISPF to interpret CICS system dump online. That, however, is beyond the scope of this book.

12.5.5 CEDF for CICS Programs

CEDF works in the same way as in early versions of CICS, provided the program is defined with CEDF(YES), which is the default.

12.5.6 EXEC CICS HANDLE CONDITION

You can use this to transfer control in a program to a user′s own code to handle errors. If you do, however, you do not get a dump for the CICS abend.

12.5.7 CICS Trace

CICS trace Level 3 provides a large amount of diagnostic data, and users may well find that much of it is not needed to read the dump. Beware of using Level 3 or 4 for the SM trace, as both levels significantly degrade system performance. However, if your user has corrupted storage, you may be forced to invoke them.

12.5.8 ON EXCEPTION Phrase

If an application program is calling a subroutine that cannot be loaded, the program can transfer control to an error-handling routine by using the ON EXCEPTION phrase of the COBOL CALL statement. However, this results in suppressing the CICS transaction dump, so you must be sure that the user does not want the dump produced.

12.5.9 System Recovery Table

You can code these to handle certain types of CICS system abends. In general, they are not very useful for application abend handling, apart from ASRA and ASRB abends.

12.5.10 Slip Traps

Code these only under instruction from IBM support staff.

12.5.11 Dump Processing

CICS/TS does not write storage accounting area data to the first eight and last eight bytes of each storage area.

Chapter 13. Performance Monitoring

We include a brief description about performance monitoring, making readers aware that the topic exists. The topic can be a book by itself. Actually, many ITSO redbook have been published on this subject. *IMS/ESA V5 Performance Guide, SG24-4637* is one just published recently.

13.1 Using IMS Display Commands

You get no useful statistical information from IMS display commands with DBCTL. IMS/DC provides many useful statistical data, but they are not available to, or in most cases, are not relevant to IMS DBCTL.

13.2 Using the IMS Monitor

The IMS MONITOR command is the quickest way to obtain basic statistical information about this online region.

13.2.1 Starting the Monitor

The IMS Monitor is activated by issuing the IMS command

```
/TRA SET ON MONITOR ALL.
```

and is turned off with the command

```
/TRA SET OFF MONITOR.
```

This starts the IMS monitor for program calls to DL/I and for scheduling of PSBs.

13.2.2 Writing the Monitor Data

The information is written to the IMS Monitor data set, IMSMON, which is normally allocated dynamically, and is written to tape or disk when monitoring is activated. The data set is not normally catalogued to MVS. However, you can catalog one if you need to. To specify the details of the IMSMON data set, use dynamic allocation macro statements. For example,

```
* Source is found in member IMS.SJIMSD.DALSRC(IMSMON)

    &SYS1  SETC 'SJIMSD'
    DFSMDA TYPE=INITIAL
    *
    * DC MONITOR DATASET
    *
           DFSMDA TYPE=DFSDCMON,DDNAME=IMSMON,
                  UNIT=3390,BUFNO=9,BLKSIZE=4096,
                  DSNAME=IMS.&SYS1..IMSMON
           DFSMDA TYPE=FINAL
           END
```

This must then be assembled and linked into USERLIB, using a job like that in Figure 128 on page 246

13.2.3 Print the Monitor Data

To print out the monitor data, you must first switch the trace off. Then run the job shown in Figure 122 to print out all the reports.

```
* JCL in member IMS.SJIMSD.JOBS(IMSMON)

//IMSDMONP JOB (999,POK),
//         CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1),
//         NOTIFY=IMSVSE1
//*
//S1      EXEC PGM=DFSUTR20,REGION=512K
//STEPLIB   DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//          DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD DISP=SHR,DSN=IMS.SJIMSD.IMSMON, *look up volser number
//          UNIT=3390,VOL=SER=......           *in JES2 output of IMS
//*                                            *control region.
//ANALYSIS  DD *
DLI CALL REPORT
DISTRIBUTION
/*
//
```

Figure 122. Sample JCL to Print IMS Monitor Data

If you want to be more selective, consult the manual *IMS/ESA, Administration Guide: System, SC26-8730*.

13.2.4 What Reports Tell

The reports from the monitor include:

- Database buffer reports
- VSAM buffer reports
- Latch conflict reports
- Region IWAITs
- Program reports by region
- Program summary
- Program I/O
- A number of DC reports that are empty
- DL/1 call summary
- Distribution appendix

Some sample output is shown in Appendix K, “IMS Monitor Reports” on page 301.

13.3 Using IMS Performance Analysis and Reporting System

You can use IMSPARS, Program 5798-CQP, to generate reports on the content of IMS system logs and obtain additional explanation of the log record types. The reports present information in the following categories:

- Transaction transit times
- IMS resource usage
- IMS resource availability

Summary reports show graphs of:

- Transaction timings
- Analysis reports
- Detailed reports

that trace individual database change activity.

Samples of the use of IMSPARS can be seen in Appendix L, "IMSPARS Reports" on page 315.

13.4 Using IMS Performance Analyzer

In general, the IMS Performance Analyzer (IMS PA) produces reports using IMS logs (SLDS) and IMSMON files as input. It has several benefits as compared with IMSPARS.

There are various IBM programs exist that do a part of what IMS PA does. However, no other IBM program does all. The IMS product utilities do some limited log analysis. For example,

- The IMS Log Transaction Analysis utility (DFSILTA0) provides response times for each transaction processed within a given time period. However, statistical analysis and trends are not easily derived from DFSILTA0's report format.
- The Statistical Analysis utility can provides statistical summaries, but requires extensive sorting. Most installations have neither the time nor the tape drives required for the statistical summaries.

Several specialized FDPs exist, primarily for transaction response reporting, and many installations have written their own log processors.

IMS PA addresses the need for a single, comprehensive log analysis program that requires a minimum of sorting and system resources. With IMS PA, the log is passed once, and there is no sorting.

Additionally, IMS PA provides online graphs of selected log data, and has facilities for exporting this data for manipulation by external programs, or for unloading to a PC.

In the event of an abend in the batch report processing or an I/O error in an input data set, IMS PA produces partial reports. Other programs may terminate with no generated output. See *IMS/ESA, Performance Analyzer User Guide, SC26-9088* for more information.

13.5 CICS Monitoring and Monitor Control Table

By default all CICS event classes are monitored. You need only code an MCT if you need to exclude some classes. You may also want to set some CICS start-up parameters to monitor the CICS system more fully, for example:

MN=ON|OFF Is monitoring On or Off?

MNCONV=NO|YES Monitoring the conversational tasks separately?

MNEVE=ON|OFF SYSEVENT monitoring active?

MNEXC=ON|OFF Exception monitoring class active?

MNFREQ=0|hhmmss Frequency of producing transaction class records for the same transaction

MNPER=ON|OFF Monitoring performance class active?

MNSYNC=NO|YES Produce a transaction class record at sync point?

MNTIME=Local|GMT Statistical records stamped in local time or GMT?

The command CEMT S MONITOR can turn monitoring on or off, and reset the monitoring frequency. See *CICS Transaction Server for OS/390 CICS Performance Guide, SC33-1699* for details.

13.6 CICS Statistics

CICS statistics are no longer automatically printed out in the CICS job log. Instead, they are stored in the MVS SMF data sets and must be unloaded from there, then printed. A typical job to achieve this could be as shown in Figure 123.

```
* JCL in member CICS.SJCICSD.DBDC.JCL(STATS)

//STUP     JOB accounting info,CLASS=A,MSGCLASS=A
//*
//*********************************************************************
//*  Step 1: Unload data from the SMF data sets
//*********************************************************************
//SMFDUMP  EXEC PGM=IFASMFDP
//INDD1     DD DSN=SYS1.MANX,DISP=SHR,AMP=('BUFSP=65536')
//INDD2     DD DSN=SYS1.MANY,DISP=SHRMON,    *look up volser number
//OUTDD1    DD DSN=user.SMF.DATA,DISP=(NEW,CATLG),
//             SPACE=(CYL,(50,10)),UNIT=SYSDA
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
       INDD(INDD1,OPTIONS(DUMP))
       INDD(INDD2,OPTIONS(DUMP))
       OUTDD(OUTDD1,TYPE(0:255))
```

Figure 123 (Part 1 of 2). Sample JCL to Print CICS Statistics from MVS SMF Data Sets

```
/*
//*****************************************************************
//*  Step 2: Optionally, select and copy the statistics records only
//*          to a journal-type data set using the journal utility
//*****************************************************************
//JUP  EXEC PGM=DFHJUP
//STEPLIB   DD DSN=CICSTS12.CICS.SDFHLOAD,DISP=SHR
//SYSUT1    DD DSN=user.SMF.DATA,DISP=SHR
//SYSUT4    DD DSN=&&SMFSTATS,DISP=(NEW,PASS),UNIT=SYSDA,
//             SPACE=(CYL,(30,10))
//SYSPRINT DD  SYSOUT=*
//*  Note that the SCSCPAAD below is your APPLID.
//SYSIN     DD *
OPTION   COPY  OFFSET=6,FLDTYP=X,VALUE=6E,FLDLEN=1,COND=M
OPTION   COPY  OFFSET=23,FLDTYP=X,VALUE=0002,FLDLEN=2,COND=M
OPTION   COPY  OFFSET=47,FLDTYP=C,VALUE=SCSCPAAD,FLDLEN=8,COND=E
/*
//*
//*****************************************************************
//*  Step 3: Sort, format and print the statistics records
//*****************************************************************
//STUP1    EXEC PGM=DFHSTUP,REGION=32M
//STEPLIB   DD DSN=CICSTS12.CICS.SDFHLOAD,DISP=SHR
//          DD DSN=CICSTS12.CICS.SDFHAUTH,DISP=SHR
//DFHSTATS  DD DSN=&&SMFSTATS,DISP=(OLD,DELETE)
//*DFHSTATS DD DSN=user.SMF.DATA,DISP=SHR
//DFHSTWRK  DD UNIT=SYSDA,SPACE=(CYL,(8,4))
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK05  DD UNIT=SYSDA,SPACE=(CYL,(4))
//DFHPRINT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSABEND  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSIN     DD *
SELECT APPLID=(SCSCPAAD)
COLLECTION TYPE=ALL
/*
//
```

Figure 123 (Part 2 of 2). Sample JCL to Print CICS Statistics from MVS SMF Data Sets

The DL/I statistics are not printed in this printout; they are printed from the IMS monitor.

13.7 MVS Workload Manager

By far the easiest way to balance the system load is to let the MVS workload manager do it. All you have to do is to define a few key goals initially in MVS, such as,

- Subsystem type
- Job or STC name (optional)

- Response time — x% less than n.milliseconds
- By user ID (optional)
- Importance of this goal (optional)
- Time of day (optional)

Code these in an IEAICSxx member of SYS1.PARMLIB. You can have one service policy for one CICS region, share the same policy between different CICS regions, or have different service policies for the same CICS region at different times of day. The MVS workload manager attempts to meet user goals as best it can. See Appendix J, “MVS Workload Manager” on page 297 for an example.

Chapter 14. SYSPLEX Considerations

If you are migrating from an environment with multiple VSE systems to an OS/390 environment with multiple MVS images, you may want to link them into a SYSPLEX.

This chapter attempts to give a brief list of those things you may want to consider.

14.1 IMS DBCTL Placement

With IMS/ESA V6, an IMS database can be shared between IMS regions. Locking at the record level allows you to run the same ACB in two IMS DBCTLs in different MVS images, provided they are not accessing the same data, and depending on the SHARELVL parameter set when the database was registered to DBRC.

The only restrictions are these:

- IMSID should be unique across a sysplex.
- You must run IRLM to control locking in all MVS images.
- The IMS DBCTL must run in the same MVS image as the CICS region or BMP that is trying to access it.

14.2 CICS VTAM ID

VTAM allows the same APPLID to be used concurrently in different MVS images. However only one of these image can be visible outside its own VTAM domain. Consequently, if you want to have several CICS regions available in different MVS images, either for fallback or for load balancing, you must give them different VTAM APPLIIDs. This is achieved by coding wild cards (*) in the VTAM APPL name, and using the GRNAME= instead of APPLID= in the SIT start-up parameter when bringing up the CICS. Thus, identical copies of a CICS terminal-owning region can be started in many parallel MVS images.

14.3 CICS Multiple Application-Owning Regions

When you define a program or transaction as remote in CICS, you must cite the SYSIDENT. If you want the SYSIDENT to be decided dynamically, depending on, for example, the workload on each CICS AOR or on the number of requests users have already routed there, you can then write a dynamic transaction routing exit to achieve this.

14.4 Links between CICSs in Different MVS Images

These must be ISC links. MRO links are allowed between CICSs only in the same MVS image.

14.5 Logging

CICS can use a coupling facility to record the log streams. If users have a cluster of application regions, then this would be the preferred way to do it, to allow different updates to the same files to be held together.

IMS keeps separate sets of logs for each DBCTL control region. You must use DBRC to run regular Change Accumulation jobs to produce merged Change Accumulation logs using input from the logs from all the DBCTLs that have participated in data sharing. See *IMS/ESA, Operations Guide, SC26-8741* for details.

Appendix A. Migration Planning Worksheets

Here are some worksheets you may find useful when analyzing application system for migration projects. The worksheets are these:

- **Planning worksheet**. Shown in Table 23 on page 230 is a generalized migration plan project worksheet for reference. Add or delete tasks as required, depending on the complexity of the project.
- **Project team worksheet**. Table 24 on page 234 shows a *who-does-what* worksheet to list the major responsibility of each team member and the associated end result.
- **Data inventory worksheet**
- **Program inventory worksheet**

Table 25 on page 236 through Table 26 on page 237 show the suggested project tasks and the suggested resources.

Table 23 (Page 1 of 4). Worksheet 1: Planning Worksheet

Milestone	Plan Date	Delivery/Criteria	Task List	Check point	Skill	Who
Migration plan completed/reviewed and approved		Fill Worksheet 2	1. Form migration team			Project leader
			2. Define project objectives and each of the team member responsibilities.			
			3. Identify migration tasks and methods.			
			4. Identify tools needed and availability.			
			5. Develop migration plan.			
			6 .Review with management and obtain approval.			
VSE software inventory completed		Fill Worksheets 3 and 4	1. Identify application systems for migration		VSE and OS/390	Application Programmer
			2. Identify what production data need migrated and location/how move to OS/390			
			3. Identify programs/where/how move to OS/390			
Essential training on OS/390		Eligible OS/390 skilled person	Take training courses from IBM			
Necessary hardware installed		Hardware no bottleneck	1. Capacity Planning (CPU/Memory/DASD/Tape/Network)			Vendor and project leader
			2. Write a new hardware proposal if you need more capacity			
			3. Purchasing			
			4. Installation			
Preinstallation Planning		1. Planning report	1. Storage Planning (Virtual storage/DASD...) according to application requirement		OS/390 RCAF	System programmer
		2. Filling Worksheet 5	2. Make security policy (Security Plan)			
			3 .Devise system standard specification (naming convention) (Program/File/Network)			

Table 23 (Page 2 of 4). Worksheet 1: Planning Worksheet

Milestone	Plan Date	Delivery/Criteria	Task List	Check point	Skill	Who
OS/390 system products installed and customized		1. OS/390 up and running	1. OS/390 ServerPac installation		OS/390 SMP/E	System Programmer
		2. Installation report	2. Customization of RACF/JES/ISPF/VTAM/etc.			
			3. DFSMS implementation if needed			
			4. CICS environment setup			
			5. IMS/DBCTL setup			
			6. Language environment setup			
			7. Network environment setup			
			8. System testing run			
VSE system migrate to OS/390 environment (not application related, see SG24-2043)		1. System environment conversion completed	1. POWER to JES migration (operation/NJE/user exit...)		VSE & OS/390	System programmer
		2. Document new implementation and definition	2. ICCF to ISPF migration (ICCF library transfer to OS/390 ISPF PDS & operation)			
		3. New operation document	3. VTAM & NCP network definitions convert to OS/390 VTAM/NCP			
			4. File system convert from VSE/VSAM, SAM to OS/390 file system (catalog...)			
			5. VSE utility converted to OS/390 utility (DFSORT/DITTO/BACKUP/RESTORE/etc.)			
Pilot application conversion completed (application only, see Chapter 5, "Application Environment" on page 33)		1. Migration environment ready	1. Identify pilot application subsystem/inventory (both online and batch)		CICS VSE & CICS TS	Application Programmer

Table 23 (Page 3 of 4). Worksheet 1: Planning Worksheet

Milestone	Plan Date	Delivery/Criteria	Task List	Check point	Skill	Who
		2. Migration methods tested	2 .Migration environment setup or verify		DL/I IMS DBCTL	
		3. Pilot subsystem migrated and tested	3. CICS/VSE system to CICS/TS system convert (system definition and files)		COBOL CICS coding	
		4. Document each conversion method	4. DL/I to IMS DBCTL conversion system definition and DB file migration			
		5. Document problems and solution	5. VSE JCL to OS/390 JCL conversion			
		6. Document operation	6. Macro level coding to command-level coding conversion on OS/390			
		7. Application backup/recovery procedure	7. DOS/COBOL or COBOL II to COBOL for VSE conversion on VSE			
			8. COBOL for VSE to COBOL for OS/390 conversion			
			9. DB management development (DB image/LOG regular backup/recovery procedure)			
			10. Develop and document operation procedure in OS/390			
			11. SAFE code recompile under OS/390			
Each application successfully migrated			Repeat some of the above steps for every application system			Application Programmer
Testing		1. Find error or make system failure	1. Set up network connection that is easy to switch over between VSE & OS/390			
		2. Meet performance requirement	2. Individual application accurate testing locally (correct business process/correct output)			

Table 23 (Page 4 of 4). Worksheet 1: Planning Worksheet

Milestone	Plan Date	Delivery/Criteria	Task List	Check point	Skill	Who
		3. Testing report	3. Integrated testing of application system with network environment (daily/monthly/annually based)			
			4. Stress testing			
			5. System/DB/Application recovery testing			
Operator training		1. Well documented operation workbook	1. OS/390 system startup/stop training			Project leader
			2. JES operation training			
			3. Network management training			
			4. Application start/stop training (CICS TS, DBCTL, and so on.)			
			5. Daily backup training			
			6. Application recovery training			
Production cut over		OS/390 system put into production	1. Make rollback plan			Project leader
			2. Transfer production data from VSE to OS/390			
			3. Switch network infrastructure from VSE to OS/390			

Table 24 (Page 1 of 2). Worksheet 2: Project Team Worksheet

Title	Responsibility	Delivery	Name
Project leader	1. Manage and control the migration project	1. Migration planning	
	2. Coordinate between migration team member and others	2. Whole migration procedure	
	3. Make migration specifications		
	4. Design migration procedures		
	5. Track migration schedule and assign necessary resource		
	6. Choose conversion tools		
	7. Plan and prepare for OS/390 production cut over		
System programmer	1. Help to design the specifications for the migration	1. System migration procedure(POWER to JES, ICCF to ISPF,...)	
	2. Help to design migration procedures for system part	2. OS/390 installation plan and report	
	3. Install and customize migration tools	3. OS/390 system operation workbook	
	4. Install, customize and migrate all related subsystems from VSE to OS/390	4. System migration problem encountered and solving techniques.	
	5. Analyze and solve conversion problem		
	6. Prepare OS/390 production system environment		
	7. Assist with OS/390 operation		
Application programmer	1. Help to design the specification of migration	1. Application migration procedure	
	2. Analyze and prepare the VSE inventory	2. Prepare JCL, convert sample macro-level to command-level programs and old COBOL to COBOL for OS/390 programs	
	3. Develop application migration procedure and select migration tools	3. Application on OS/390 daily operation workbook	
	4. Analyze and solve migration problem		
	5. Lead programs and data files migration task		
Data base administrator	1. Help to design DB migration specification	1. DB backup and recovery procedure	
	2. New DB planning and implementation based on DL/I DB	2. DL/I to IMS DBCTL migration procedure (both program and DB data part)	

Table 24 (Page 2 of 2). Worksheet 2: Project Team Worksheet			
Title	**Responsibility**	**Delivery**	**Name**
	3. Design DB data/program migration procedure (both batch and online)		
	4. Design DB backup/recovery procedure		
	5. Help to set up DB testing data		
	6. Help to make cut-over rollback plan		
Operator	1. Perform integrated and stress testing and document the problem		
	2. Implement new OS/390 operation procedure		
	3. Help cut over to production		

Table 25. Worksheet 3: Application System Programs Inventory

VSE Program Source			**Access Resource**			**JCL Source**	
Transaction	Program	Language (ICCF Library)	VSAM File	DL/I	SAM/Tape	Compiler	Running

Table 26. Worksheet 4: Application System Data Inventory

File/DB File	VSAM Define JCL		DL/I Related Definition			File Initial JCL
	ICCF Library	VSAM Catalog	DBD Definition	PCB Definition	ACB Definition	

Appendix B. Resources Needed by Initial Installer

This appendix lists items from Chapter 7, “System Environment Customization” on page 57 that users need only for the very first initial installation, which on many sites will have been done by others, or by the service provider.

Many of the JCL samples listed in this Appendix can be found either in IMS or CICS TS product publications. Some are generated by the installation processing. We quote here some commonly used jobs that can be used repeatedly. Users from a VSE environment who are new to OS/390 are likely to find it useful. However, as many of these jobs have been customized for the ITSO environment, users may need to make adjustments for their own environment.

B.1 Software Levels on OS/390

OS/390 V 2 should be at Release 4 or higher.

CICS TS V 1 should be at Release 2 or higher.

IMS/ESA Version 6.1 should have PTF UQ13783 applied and accepted.

B.2 System Data Sets

We give examples here for IMS/ESA and CICS TS system environment only. For the OS/390 system environment, consult with OS/390 publications for detailed information.

B.2.1 SMP/E Data Sets

These are used for installation and maintenance of the CICS and IMS software. They are not used in day-to-day operations. You could archive them to tape, especially if your users do not intend to apply their own system maintenance. There many ways to set up an SMP/E environment. Some installations choose to have one set of SMP/E data sets for each major subsystem. Others may combine everything into one. Figure 124 on page 240 and Figure 125 on page 241 show ITSO environments that have separated IMS/ESA and CICS TS SMP/E environments. The listings are for reference only and may differ in other installations.

```
IMS610.ADXRLOAD
IMS610.ADXRSAMP
IMS610.DBSOURCE
IMS610.DFSCLST
IMS610.DFSCLSTA
IMS610.DFSEXEC
IMS610.DFSEXECA
IMS610.DFSISRC
IMS610.DFSISRCA
IMS610.DFSMLIB
IMS610.DFSMLIBA
IMS610.DFSPLIB
IMS610.DFSPLIBA
IMS610.DFSRTRM
IMS610.DFSRTRMA
IMS610.DFSSLIB
IMS610.DFSSLIBA
IMS610.DFSTLIB
IMS610.DZONE.CSI
IMS610.GENLIB
IMS610.GENLIBA
IMS610.GENLIBB
IMS610.GLOBAL.CSI
IMS610.INSTALIB
IMS610.INSTATBL
IMS610.LOAD
IMS610.MACLIB
IMS610.MATRIX
IMS610.MODBLKS
IMS610.OBJDSET
IMS610.OPTIONS
IMS610.PROCLIB
IMS610.PSBLIB
IMS610.REFERAL
IMS610.RESLIB
IMS610.SDXRRESL
IMS610.SDXRSAMP
IMS610.SMPLTS
IMS610.SMPMTS
IMS610.SMPPTS
IMS610.SMPSCDS
IMS610.SMPSTS
IMS610.SVSOURCE
IMS610.TMSOURCE
IMS610.TZONE.CSI
```

Figure 124. IMS SMP/E Data Sets

```
CICSTS12.ADFHAPD1
CICSTS12.ADFHAPD2
CICSTS12.ADFHCLIB
CICSTS12.ADFHCOB
CICSTS12.ADFHC370
CICSTS12.ADFHINST
CICSTS12.ADFHLANG
CICSTS12.ADFHMAC
CICSTS12.ADFHMLIB
CICSTS12.ADFHMOD
CICSTS12.ADFHMSGS
CICSTS12.ADFHMSRC
CICSTS12.ADFHPARM
CICSTS12.ADFHPLIB
CICSTS12.ADFHPL1
CICSTS12.ADFHPROC
CICSTS12.ADFHSAMP
CICSTS12.SDFHAPD1
CICSTS12.SDFHAPD2
CICSTS12.SDFHAUTH
CICSTS12.SDFHCLIB
CICSTS12.SDFHCOB
CICSTS12.SDFHC370
CICSTS12.SDFHEXCI
CICSTS12.SDFHINST
CICSTS12.SDFHLANG
CICSTS12.SDFHLINK
CICSTS12.SDFHLLIB
CICSTS12.SDFHLOAD
CICSTS12.SDFHLPA
CICSTS12.SDFHMAC
CICSTS12.SDFHMLIB
CICSTS12.SDFHMSGS
CICSTS12.SDFHMSRC
CICSTS12.SDFHPARM
CICSTS12.SDFHPLIB
CICSTS12.SDFHPL1
CICSTS12.SDFHPROC
CICSTS12.SDFHSAMP
CICSTS12.GZONE.CSI
CICSTS12.DZONE.CSI
CICSTS12.DZONE.SMPLOG
CICSTS12.SMPLTS
CICSTS12.SMPMTS
CICSTS12.SMPPTS
CICSTS12.SMPSCDS
CICSTS12.SMPSTS
CICSTS12.TDFHINST
CICSTS12.TZONE.CSI
CICSTS12.TZONE.SMPLOG
CICSTS12.XDFHINST
```

Figure 125. CICS SMP/E Data Sets

B.2.2 IDCAMS Statements

The OS/390 system installer would have executed the following IDCAMS statements defining IMS.* and CICS.* high level qualifiers:

```
DEF ALIAS(NAME('IMS')  rel('CATALOG.TOTICF1.VTOTCAT'))
DEF ALIAS(NAME('CICS') rel('CATALOG.TOTICF1.VTOTCI1'))
```

They would have set these as aliases in the OS/390 master catalog so that all the data sets are catalogued in the same OS/390 user catalogs. In the ITSO environment, the user catalogs are:

```
     CATALOG.TOTICF1.VTOTCAT
and  CATALOG.TOTICF1.VTOTCI1
```

Alternatively, the above statements could also be incorporated into MVS utility JCL and run as an MVS batch job.

B.2.3 RACF Definitions

The OS/390 system installer would have also executed the following RACF statements:

```
AG CICS OWNER(SYS1)   SUPGROUP(SYS1)
AG IMS  OWNER(SYS1)   SUPGROUP(SYS1)
AD 'CICS.**' GENERIC  OWNER(CICS) UACC(UPDATE)
AD 'IMS.**'  GENERIC  OWNER(IMS)  UACC(UPDATE)
AD 'CICS.SJCICSD.**'  GENERIC  OWNER(CICS) UACC(UPDATE)
AD 'IMS.SJIMSD.**'    GENERIC  OWNER(IMS)  UACC(UPDATE)
RDEF AIMS (BMP01)     OWNER(IMS)  UACC(READ)
```

These define IMS.* and CICS.* as RACF generic resources. You users, then, can access all CICS and IMS data sets fully. These RACF statements can be executed in many forms. They can run as an MVS batch job, or they can use RACF ISPF interactive panels system. The statements cited above are for reference only. Your user may want to modify these definitions to meet their own environment needs.

Note

To execute the above RACF statements, assume that SYS1 group already exists and that TSO user is owned by it. If the owner is called differently, change the group name SYS1 accordingly.

Also, if your users are not authorized to use RACF commands, work with the one who is responsible for OS/390 security.

RACF is IBM's security server for OS/390. Users who have a security subsystem from another vendor should consult the product publication for the environment set up.

B.2.4 IMS/ESA and CICS TS System Data Sets

In ITSO, our IMS/ESA and CICS TS system data sets are set up (see Chapter 7, "System Environment Customization" on page 57) under the names of

IMS.SJIMSD.*

CICS.SJCICSD.*

Refer to *IMS/ESA, Installation Volume I, GC26-8736* and *CICS Transaction Server for OS/390 Installation Guide, GC33-1681* for detailed information on function and usage for each data set. Users are free to use any name.

B.3 APF CICS and IMS Libraries

In SYS1.PARMLIB, you find several members called PROGxx. One contains many statements like these:

```
APF ADD
    DSNAME(IMS.SJIMSD.RESLIB)
    VOLUME(TOTIM5)
APF ADD
    DSNAME(IMS.SJCICSD.SDFHAUTH)
    VOLUME(TOTTS6)
```

In general, whether your users can create their own PROGxx statements is doubtful. Both *IMS/ESA, Installation Volume I, GC26-8736* and *CICS Transaction Server for OS/390 Installation Guide, GC33-1681* have detailed information about the authorized library requirement. Follow the example above to code the necessary statements, and once you have coded them, issue the MVS operator command at an MVS console, or through SDSF, to activate the authorized library set.

```
      SET PROG=xx
or
     /SET PROG=xx    if issued from SDSF.
```

B.4 Define IPCS Commands for CICS/TS to OS/390

To enable your users to format dumps from CICS TS region, you need to define a member, named DFHIPCSP, in OS/390 SYS1.PARMLIB. Figure 126 on page 244 shows an example of member DFHIPCSP.

```
/* ================================================================ */
/* $MAC(DFHIPCSP)                                                   */
/*                                                                  */
/* NAME: DFHIPCSP                                                   */
/*                                                                  */
/* DESCRIPTIVE NAME: CICS IPCS Parmlib IMBED Member.                */
/*                                                                  */
/*             @BANNER_START@                                       */
/*             Licensed Materials - Property of IBM                 */
/*                                                                  */
/*             "Restricted Materials of IBM"  5655-147              */
/*                                                                  */
/*             (C) Copyright IBM Corp. 1992,                        */
/*                                                                  */
/*             CICS 5.2.0                                           */
/*             (Element of CICS Transaction Server                  */
/*                         Version 1 Release 2)                     */
/*             @BANNER_END@                                         */
/*                                                                  */
/* Function: DFHIPCSP is used to define materials made available for*/
/* CICS formatter routine under IPCS.                               */
/* ---------------------------------------------------------------- */
/* CHANGE ACTIVITY:                                                 */
/*                                                                  */
/*  PN= REASON REL YYMMDD HDXIII : REMARKS                          */
/* $01  Reserved for APAR fix                                       */
/* $D0= I05493 330 900625 HD5XTY : First Coded                      */
/* $H1  Reserved for hardware support                               */
/* $L1= BS1    410 920122 HD6ACPA: Add new exit for 410             */
/* $L2= BS1    510 940224 HD6ACPA: Add new exit for 420             */
/* $L3= BS1    520 961112 HD3SCWG: Add release 520                  */
/* $P1= M90951 330 910925 HD2GJST: Change release number to 330     */
/* $P2= M85216 410 940118 HD6ACPA: Remove redundant releases        */
/* $P3= M94961 510 950724 HD6ACPA: Add release 510                  */
/* ================================================================ */
/* IPCS Verb Exits                                                  */
/* ================================================================ */
  EXIT EP(DFHPDX) VERB(CICSDATA) ABSTRACT('CICS analysis')
/* ================================================================ */
/* The following entries can be activated by changing the name of   */
/* the DFHPDX in each release to the name defined in the EP field   */
/* ================================================================ */
  EXIT EP(DFHPD212) VERB(CICS212) ABSTRACT(+
    'CICS Version 2 Release 1.2 analysis')
  EXIT EP(DFHPD321) VERB(CICS321) ABSTRACT(+
    'CICS Version 3 Release 2.1 analysis')
  EXIT EP(DFHPD330) VERB(CICS330) ABSTRACT(+
    'CICS Version 3 Release 3 analysis')
  EXIT EP(DFHPD410) VERB(CICS410) ABSTRACT(+
    'CICS Version 4 Release 1 analysis')
  EXIT EP(DFHPD510) VERB(CICS510) ABSTRACT(+
    'CICS Transaction Server for OS/390 Release 1 analysis')
  EXIT EP(DFHPD520) VERB(CICS520) ABSTRACT(+
    'CICS Transaction Server for OS/390 Release 2 analysis')
```

Figure 126. SYS1.PARMLIB Member DFHIPCSP

The example member in Figure 126 shows all possible current available releases of MVS CICS and OS/390 CICS TS.

B.5 Define VTAM APPLID for CICS.

Include the following statements in a member of the library of SYS1.VTAMLST.

```
<label>  VBUILD TYPE=APPL
SCSCPAAD APPL AUTH=(ACQ,VPACE,PASS),VPACING=0,EAS=5000,PARSESS=YES,    X
               SONSCIP=YES,APPC=NO
```

Once the above VTAM definition has been coded, issue the operator command to activate the VTAM application:

```
       V NET,ACT,ID=APCPAAD,SCOPE=ALL
```

You can also include this member name in the VTAM start-up list ATCCON00, so that it is always available to your users.

B.6 Define Dynamic Allocations for IMS

Dynamic allocation is a convenient way to manage DBCTL operation. It also ensures full-time system availability (24 hours x 7 days). The IMS macro DFSMDA should be used to inform DBCTL about system and application resources. DBCTL then will be able to find the data sets dynamically when it references the resources.

Figure 127 shows an example of a typical dynamic allocation source.

```
&SYS1    SETC 'SJIMSD'
         DFSMDA TYPE=INITIAL
*
* OLDS, 3 PRIMARY AND 3 SECONDARY
*
  DFSMDA TYPE=OLDS,DDNAME=DFSOLS00,DISP=SHR,DSNAME=IMS.&SYS1..OLS00
  DFSMDA TYPE=OLDS,DDNAME=DFSOLP00,DISP=SHR,DSNAME=IMS.&SYS1..OLP00
  DFSMDA TYPE=OLDS,DDNAME=DFSOLS01,DISP=SHR,DSNAME=IMS.&SYS1..OLS01
  DFSMDA TYPE=OLDS,DDNAME=DFSOLP01,DISP=SHR,DSNAME=IMS.&SYS1..OLP01
  DFSMDA TYPE=OLDS,DDNAME=DFSOLS02,DISP=SHR,DSNAME=IMS.&SYS1..OLS02
  DFSMDA TYPE=OLDS,DDNAME=DFSOLP02,DISP=SHR,DSNAME=IMS.&SYS1..OLP02
*
  DFSMDA TYPE=OLDS,DDNAME=DFSOLS03,DISP=SHR,DSNAME=IMS.&SYS1..OLS03
  DFSMDA TYPE=OLDS,DDNAME=DFSOLP03,DISP=SHR,DSNAME=IMS.&SYS1..OLP03
*
         DFSMDA TYPE=FINAL
         END
```

Figure 127. Dynamic Allocation Macro Statements for IMS Log Data Sets

Once the dynamic allocation source is coded, Figure 128 on page 246 shows JCL that can be used to assemble the macros and store the load module in a library. The library should be APF-authorized and available to DBCTL regions (STEPLIB or LINKLST).

```
*   JCL placed in IMS.SJIMSD.JOBS(DYNALLOC)

//IMSDALOC PROC SOUT='*',SYS1='SJIMSD.'
//ASSEM    EXEC PGM=ASMA90,PARM='ALIGN,DECK,NOOBJECT'
//SYSLIB     DD DSN=IMS.&SYS1.MACLIB,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(25,10))
//SYSPUNCH   DD DSN=&&OBJMOD,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//           SPACE=(TRK,(10,10)),UNIT=SYSDA,DISP=(NEW,PASS)
//SYSPRINT   DD SYSOUT=&SOUT
//*
//BLDMBR   EXEC PGM=IEBUPDTE,PARM='NEW',COND=(7,LT,ASSEM)
//SYSPRINT   DD SYSOUT=&SOUT
//SYSUT2     DD DSN=&&TEMPPDS,UNIT=SYSDA,DISP=(NEW,PASS,DELETE),
//           SPACE=(TRK,(10,5,10)),DCB=(RECFM=FB,LRECL=80)
//SYSIN      DD DSN=*.ASSEM.SYSPUNCH,DISP=(OLD,DELETE,DELETE)
//*
//LNKEDT   EXEC PGM=IEWL,PARM='LIST,XREF,LET',
//           COND=((7,LT,ASSEM),(3,LT,BLDMBR))
//SYSUT1     DD UNIT=SYSDA,SPACE=(1024,(100,50))
//SYSLIB     DD DUMMY
//SYSPRINT   DD SYSOUT=&SOUT
//SYSLMOD    DD DSN=IMS.&SYS1.USERLIB,DISP=SHR
//OBJMOD     DD DSN=&&TEMPPDS,DISP=(OLD,PASS)
//SYSLIN     DD DSN=&&TEMPPDS(LNKCTL),DISP=(OLD,DELETE,DELETE),
//           VOL=REF=*.OBJMOD
//       PEND
//*
//       SET SYS1='SJIMSD.'
//MAKE   EXEC IMSDALOC,SYS1=&SYS1
//SYSIN    DD DISP=SHR,DSN=IMS.&SYS1.DALSRC(IMSOS390)     **
//
```

Figure 128. Sample JCL for Assembling Dynamic Allocation Macro Statements

IMS installation procedures create completed dynamics allocation jobs for all IMS system resources and for the included IVP application. See *IMS/ESA, Installation Volume I, GC26-8736* for detailed information.

B.7 Creation of Parameter Modules

Create OS/390 CICS TS online region table modules is no different from creating those in VSE CICS, except for the execution JCL. The JCL is standard practice for CICS system programmers; however, for those who are new to OS/390, getting the right syntax for OS/390 JCL may take time. We, therefore, list JCL for some of the major system tables for reference. All JCLs are used for this residency project.

B.7.1 Creating CICS SIT Module

Figure 129 shows the CICS TS online region system initialization table assembly job.

```
* JCL placed in CICS.SJCICSD.DBDC.JCL(SITASM)

//SITASM   PROC
//C        EXEC PGM=ASMA90,PARM='OBJ,NODECK,LIST',REGION=4M
//SYSPRINT   DD SYSOUT=*,DCB=(RECFM=FB,LRECL=121,BLKSIZE=0)
//SYSLIB     DD DISP=SHR,DSN=CICSTS12.CICS.SDFHMAC
//SYSLIN     DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,BUFNO=40)
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//*
//* Link the SIT.
//*
//L        EXEC PGM=IEWL,PARM='LIST,NE,OL',REGION=4M
//SYSPRINT   DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=0)
//SYSLMOD    DD DISP=SHR,DSN=CICS.SJCICSD.SDFHAUTH
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN     DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//       PEND
//*
//*
//       EXEC PROC=SITASM
//C.SYSIN  DD *
        DFHSIT TYPE=CSECT,
               SUFFIX=D0,
               SYSIDNT=CICD
        END    DFHSITBA
/*
//L.SYSLIN DD
//         DD *
        NAME DFHSITD0(R)
/*
//
```

Figure 129. Sample JCL for Assembling Default SIT Module for CICS TS

The load module of the SIT is the only table you need to place in the OS/390 authorized library.

B.7.2 Creating CICS Start-up PLT Module

Not every CICS/TS online region requires a PLT. Figure 130 shows the execution JCL to assemble one if needed for reference.

```
JCL placed in CICS.SJCICSD.DBDC.JCL(PLTASM)

//PLTASM   PROC
//C        EXEC PGM=ASMA90,PARM='OBJ,NODECK,LIST',REGION=4M
//SYSPRINT   DD SYSOUT=*,DCB=(RECFM=FB,LRECL=121,BLKSIZE=0)
//SYSLIB     DD DISP=SHR,DSN=CICSTS12.CICS.SDFHMAC
//SYSLIN     DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,BUFNO=40)
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//*
//* Link the PLT.
//*
//L        EXEC PGM=IEWL,PARM='LIST,NCAL,LET',REGION=4M
//SYSPRINT   DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=0)
//SYSLMOD    DD DISP=SHR,DSN=CICS.SJCICSD.TABLOAD
//SYSUT1     DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN     DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//       PEND
//*
//*
//       EXEC PROC=PLTASM
//C.SYSIN  DD DISP=SHR,DSN=CICS.SJCICSD.PARMLIB(PLTPI)
/*
//L.SYSLIN DD
//         DD *
        NAME DFHPLTPI(R)
/*
//
```

Figure 130. JCL for Assembling Default SIT Module for CICS TS

The same JCL shown in Figure 130 can also be used to assemble most of the other CICS/TS tables. Replace source input and change the load module name as required.

B.7.3 Creating CICS-IMS DRA Module

Assembling the DRA load module is similar to other CICS/TS table modules. However, its source macro is stored on the IMS system macro library. You need to add the IMS RESLIB in the execution JCL and place the load module in an MVS-authorized library. Figure 131 shows the sample JCL.

```
* JCL placed in CICS.SJCICSD.DBDC.JCL(DRAASM)

//DRAASM PROC
//C      EXEC PGM=ASMA90,PARM='OBJ,NODECK,LIST,ALIGN',REGION=4M
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=121,BLKSIZE=0)
//SYSLIB   DD DISP=SHR,DSN=CICSTS12.CICS.SDFHMAC
//         DD DISP=SHR,DSN=IMS.SJIMSD.MACLIB
//SYSLIN   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,BUFNO=40)
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//*
//* Link the module.
//*
//L      EXEC PGM=IEWL,PARM='LIST,LET,NCAL',REGION=4M
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=0)
//SYSLMOD  DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN   DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//       PEND
//*
//*
//       EXEC PROC=DRAASM
//C.SYSIN  DD *
        DFSPRP  DSECT=NO,
              DSNAME=IMS.SJIMSD.RESLIB,
              DDNAME=CCTLDD,
              DBCTLID=IMSD,
              MINTHRD=003,
              MAXTHRD=099,
              SOD=3,
              AGN=BMP01
         END
/*
//L.SYSLIN DD
//         DD *
        NAME DFSPZP0D(R)
/*
//
```

Figure 131. JCL for Assembling DRA Module for CICS TS

B.8 The IMS DBCTL Full Generation Processing

This process runs against the SMP data sets, and is used to define resources to IMS, and to include the latest maintenance. A typical process procedure is shown in the following sections.

B.8.1 IMS DBCTL Full Generation Source

IMS full-generation source is created by the IMS installation processing. In general, an installation needs to run a full generation once only, unless indicated in system maintenance PTFs. The full generation source may require some customization depending on user environment. We recommend that you generate all possible combinations of the entire enterprise environment in the full generation to avoid missing pieces. You can customize each each individual DBCTL region as needed later. Figure 132 shows the full generation source.

```
* Place statements in member IMS.SJIMSD.STAGE1.SOURCE(FULLGEN)


***********************************************************************
* INSTALL/IVP IMS 6.1
*
* SKELETON: DFSIXSC1
*
* FUNCTION: STAGE 1 SOURCE FOR A DBT SYSTEM
***********************************************************************
*
**************************************************************@SCPYRT**
*                                                                     *
*         Licensed Materials - Property of IBM                        *
*                                                                     *
*         "Restricted Materials of IBM"                               *
*                                                                     *
*         5655-158 (C) Copyright IBM Corp. 1989                       *
*         All Rights Reserved.                                        *
*                                                                     *
*         US Government Users Restricted Rights -                     *
*         Use, duplication or disclosure restricted by                *
*         GSA ADP schedule contract with IBM Corp.                    *
*                                                                     *
**************************************************************@ECPYRT**
*
* IMSCTRL MACRO --
*
         IMSCTRL  SYSTEM=(VS/2,(ALL,DBCTL),390),                       X
               IRLM=YES,                                               X
               IRLMNM=IRLM,                                            X
               CMDCHAR=>,                                              X
               DBRC=(YES,YES),                                         X
               DBRCNM=DBRC,   ,                                        X
               DLINM=DLISAS,                                           X
               IMSID=IMSD,                                             X
               MAXIO=(,015),                                           X
               MAXREGN=(005,512K,A,A),                                 X
               MCS=(2,7),                                              X
               NAMECHK=YES
```

Figure 132 (Part 1 of 4). IMS DBCTL Full-Generation Source

```
*
* IMSCTF  MACRO --
*
         IMSCTF SVCNO=(,203,202),                                      X
               LOG=DUAL,                                               X
               CPLOG=3000,                                             X
               RDS=(3390,4096),                                        X
               PRDR=IMSDRDR
*
* FPCTRL   MACRO --
*
*        FPCTRL OTHREAD=5,
*              BFALLOC=(10,50,2048)
*
* BUFPOOLS MACRO --
*
         BUFPOOLS PSB=24000,                                           X
               SASPSB=(4000,20000),                                    X
               PSBW=12000,                                             X
               DMB=24000,                                              X
               FORMAT=(24000,256),                                     X
               FRE=30
*
* SECURITY MACRO --
*
         SECURITY RCLASS=IMS,TYPE=(RACFAGN)
*
*        DBCTL LINK TO CICS PROGRAM FOR COMMAND INTERFACE
*
         APPLCTN PSB=DFHDBMP,SCHDTYP=PARALLEL,PGMTYPE=(BATCH)
*
**********************************************************************
*   IVP DATABASES DEFINITION
**********************************************************************
         DATABASE DBD=IVPDB1,ACCESS=UP                HIDAM/OSAM
         DATABASE INDEX,DBD=IVPDB1I,ACCESS=UP         HIDAM/VSAM INDEX
         DATABASE DBD=IVPDB2,ACCESS=UP                HDAM/VSAM
*        DATABASE DBD=IVPDB3,ACCESS=UP                DEDB
*        DATABASE DBD=IVPDB4                          MSDB
**********************************************************************
*   IVP BATCH/BMP APPLICATION DEFINITION
**********************************************************************
         SPACE 2
         APPLCTN PSB=DFSIVP6,PGMTYPE=BATCH            HIDAM/OSAM-ASSEM
         APPLCTN PSB=DFSIVP61,PGMTYPE=BATCH           HIDAM/OSAM-PASCAL
         APPLCTN PSB=DFSIVP62,PGMTYPE=BATCH           HIDAM/OSAM-C
         APPLCTN PSB=DFSIVP64,PGMTYPE=BATCH           HIDAM/OSAM-COBOL
         APPLCTN PSB=DFSIVP65,PGMTYPE=BATCH           HIDAM/OSAM-REXX
         APPLCTN PSB=DFSIVP7,PGMTYPE=BATCH            HDAM/VSAM
         APPLCTN PSB=DFSIVP8,PGMTYPE=BATCH            DEDB/VSAM
         APPLCTN PSB=DFSIVP9,PGMTYPE=BATCH            HIDAM/OSAM OLIC
         APPLCTN PSB=DFSIVPA,PGMTYPE=BATCH            HIDAM LOAD
         APPLCTN PSB=DFSIVPB,PGMTYPE=BATCH            HDAM  LOAD
*        APPLCTN PSB=DFSIVPC,PGMTYPE=BATCH            DEDB (DB LOAD)
         SPACE 2
**********************************************************************
*   IMS SAMPLE DATABASES DEFINITION
**********************************************************************
         SPACE 2
         DATABASE DBD=DI21PART,ACCESS=UP               HISAM/VSAM
         EJECT  ,
```

Figure 132 (Part 2 of 4). IMS DBCTL Full-Generation Source

```
**********************************************************************
*   IMS SAMPLE APPLICATION DEFINITION - CICS IVP
**********************************************************************
         SPACE 2
         APPLCTN PSB=DFHSAM04,PGMTYPE=BATCH
         APPLCTN PSB=DFHSAM14,PGMTYPE=BATCH
         APPLCTN PSB=DFHSAM24,PGMTYPE=BATCH
         APPLCTN PSB=DFHSAM05,PGMTYPE=BATCH
         APPLCTN PSB=DFHSAM15,PGMTYPE=BATCH
         APPLCTN PSB=DFHSAM25,PGMTYPE=BATCH
         EJECT  ,
**********************************************************************
*   IMS SAMPLE APPLICATION DEFINITION
**********************************************************************
         SPACE 2
         APPLCTN  PSB=DFSSAM01,PGMTYPE=BATCH
         APPLCTN  PSB=DFSSAM02
         APPLCTN  PSB=DFSSAM03
         APPLCTN  PSB=DFSSAM04
         APPLCTN  PSB=DFSSAM05
         APPLCTN  PSB=DFSSAM06
         APPLCTN  PSB=DFSSAM07
         APPLCTN  PSB=DFSSAM08,PGMTYPE=BATCH
         APPLCTN  PSB=DFSSAM09,PGMTYPE=BATCH           GENERAL PURPOSE
         SPACE 2
*******************************************************************
*   FAST PATH SAMPLE DATABASES DEFINITION
**********************************************************************
*        SPACE 2
*        DATABASE DBD=DBFSAMD1           GENERAL LEDGER - MSDB
*        DATABASE DBD=DBFSAMD2                   TELLER - MSDB
*        DATABASE DBD=DBFSAMD3,ACCESS=UP CUSTOMER ACCNT - DEDB
*        DATABASE DBD=DBFSAMD4,ACCESS=UP CUSTOMER  LOAN - HDAM/VSAM
*        EJECT  ,
**********************************************************************
*   FAST PATH SAMPLE APPLICATION DEFINITION
**********************************************************************
*        SPACE 2
*        APPLCTN  PSB=DBFSAMP1,PGMTYPE=BATCH       DEDB LOAD
*        APPLCTN  PSB=DBFSAMP2,PGMTYPE=BATCH       HDAM LOAD
*        APPLCTN  PSB=DBFSAMP4
*        APPLCTN  PSB=DBFSAMP5,PGMTYPE=BATCH       HDAM MISC.
*        APPLCTN  PSB=DBFSAMP6,PGMTYPE=BATCH       DEDB MISC.
         SPACE 2
         COPY APPLDB   - initially empty - CICS DDIR & PDIR go here.
         SPACE 2
```

Figure 132 (Part 3 of 4). IMS DBCTL Full-Generation Source

```
*
* IMSGEN MACRO --
*
         IMSGEN ASM=(HLASM,SYSLIN),ASMPRT=OFF,                         X
               LKPRT=(XREF,LIST),LKSIZE=(880K,64K),LKRGN=900K,         X
               SUFFIX=D,                                               X
               SURVEY=NO,                                              X
               TERM=YES,                                               X
               MACLIB=ALL,                                             X
               NODE=(IMS.SJIMSD,                                       X
               IMS610,                                                 X
               IMS610),                                                X
               OBJDSET=IMS.SJIMSD.OBJDSET,                             X
               PROCLIB=NO,                                             X
               USERLIB=IMS.SJIMSD.RESLIB,                              X
               UMAC0=IMS.SJIMSD.MACLIB,                                X
               SYSMAC=SYS1.MACLIB,                                     X
               MODGEN=SYS1.MODGEN,                                     X
               UMAC1=,                                                 X
               UMAC2=,                                                 X
               UMAC3=,                                                 X
               ONEJOB=(),                                              X
               JCL=(IMSGEN,                                            X
               (999,POK),                                              X
               ,T,                                                     X
               (CLASS=A,MSGLEVEL=(1,1),REGION=32M,TYPRUN=HOLD)),       X
               SCL=(8,,(TIME=600)),                                    X
               UJCL1='/*NOTIFY IMSVSE1',                               X
               UJCL2=,                                                 X
               UJCL3=,                                                 X
               UJCL4=,                                                 X
               UJCL5=
         END ,
```

Figure 132 (Part 4 of 4). IMS DBCTL Full-Generation Source

Figure 132 on page 250 also shows that we comment out all the IMS Fast-Path macros. This is because the VSE local DLI environment does not use IMS Fast-Path databases. For this project, we therefore limited our environment to only full-function databases.

B.8.2 Preprocess and Stage One

Run the job shown in Figure 133 to check the syntax of the IMSGEN source.

```
* JCL placed in IMS.SJIMSD.DBDC.JCL($2ALLGEN)

//IMSGENS1 JOB (999,POK),CLASS=A,MSGCLASS=T,
// MSGLEVEL=(1,1),NOTIFY=IMSVSE1,REGION=32M
//      SET SYS1=SJIMSD
//      SET SYSSMP=IMS610
//*
//*********************************************************************
//* INSTALL/IVP IMS 6.1
//*
//* FUNCTION: IMS SYSTEM DEFINITION PREPROCESSOR - FOR NAME VALIDATION
//************************************************************@SCPYRT**
//*                                                                  *
//*       Licensed Materials - Property of IBM                       *
//*                                                                  *
//*       "Restricted Materials of IBM"                              *
//*                                                                  *
//*       5655-158 (C) Copyright IBM Corp. 1974,1996                 *
//*       All Rights Reserved.                                       *
//*                                                                  *
//*       US Government Users Restricted Rights -                    *
//*       Use, duplication or disclosure restricted by               *
//*       GSA ADP schedule contract with IBM Corp.                   *
//*                                                                  *
//************************************************************@ECPYRT**
//*
//PREPROC  EXEC PGM=DFSPRE00,PARM='N,N'
//STEPLIB    DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//SYSSUMPR   DD SYSOUT=*
//SYSCYLPR   DD SYSOUT=*
//SYSPRINT   DD SYSOUT=*
//SYSLIB     DD DISP=SHR,DSN=IMS.&SYS1..STAGE1.SOURCE
//           DD DISP=SHR,DSN=&SYSSMP..GENLIB
//           DD DISP=SHR,DSN=&SYSSMP..GENLIBA
//           DD DISP=SHR,DSN=&SYSSMP..GENLIBB
//SYSIN80    DD DISP=(,DELETE),DSN=&&SYSIN,
//           UNIT=3390,SPACE=(CYL,(5,5)),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DSORG=PS)
//SYSUT1     DD UNIT=3390,SPACE=(CYL,(5,5))
//SYSLIN     DD DISP=(,DELETE),DSN=&&SYSLIN,
//           UNIT=3390,SPACE=(CYL,(1,5)),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DSORG=PS)
//SYSPRT80   DD DISP=(,DELETE),DSN=&&SYSPRT,
//           UNIT=3390,SPACE=(CYL,(5,5)),
//           DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3146,DSORG=PS)
//SYSCOBJ    DD DUMMY,DCB=BLKSIZE=80    STAGE1 OUTPUT DESTINATION
//SYSIN      DD DISP=SHR,
//           DSN=IMS.SJIMSD.STAGE1.SOURCE(FULLGEN)
//
```

Figure 133. Sample JCL for IMS Generation Source Verification

Once no error is found, Stage 1 generation can be continued as shown in Figure 134 on page 255.

```
* JCL placed in IMS.SJIMSD.DBDC.JCL($2ALLGEN)

//IMSGENS1 JOB (999,POK),CLASS=A,MSGCLASS=T,
// MSGLEVEL=(1,1),NOTIFY=IMSVSE1,REGION=32M
//     SET SYS1=SJIMSD
//     SET SYSSMP=IMS610
//*
//*********************************************************************
//* INSTALL/IVP IMS 6.1
//*
//* FUNCTION: RUN THE IMS SYSTEM DEFINITION STAGE1
//***********************************************************@SCPYRT**
//*                                                                  *
//*       Licensed Materials - Property of IBM                       *
//*                                                                  *
//*       "Restricted Materials of IBM"                              *
//*                                                                  *
//*       5655-158 (C) Copyright IBM Corp. 1974,1996                 *
//*       All Rights Reserved.                                       *
//*                                                                  *
//*       US Government Users Restricted Rights -                    *
//*       Use, duplication or disclosure restricted by               *
//*       GSA ADP schedule contract with IBM Corp.                   *
//*                                                                  *
//***********************************************************@ECPYRT**
//*
//STAGE1  EXEC PGM=ASMA90,PARM='NOOBJ,DECK',TIME=(600)
//SYSPRINT  DD SYSOUT=*
//SYSLIB    DD DISP=SHR,DSN=IMS.&SYS1..STAGE1.SOURCE
//          DD DISP=SHR,DSN=&SYSSMP..GENLIB
//          DD DISP=SHR,DSN=&SYSSMP..GENLIBA
//          DD DISP=SHR,DSN=&SYSSMP..GENLIBB
//SYSPUNCH  DD DISP=SHR,
//          DSN=IMS.SJIMSD.STAGE1.SOURCE(STAGE2)
//SYSUT1    DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSUT2    DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSUT3    DD UNIT=3390,SPACE=(CYL,(05,05)),DCB=OPTCD=C
//SYSIN     DD DISP=SHR,
//          DSN=IMS.SJIMSD.STAGE1.SOURCE(FULLGEN)
//
```

Figure 134. Sample JCL for IMS Stage 1 Generation

B.8.3 Stage 2

This then submits eight Stage 2 jobs, which are also saved in member IMS.SJIMSD.STAGE1.SOURCE(STAGE2). These jobs are called IMSGEN01 - IMSGEN08, and should be released manually, each one starting only when the previous one has finished. Check that all finish with zero return code, RC=0.

B.8.3.1 Run the SMP/E Steps

Although it is not compulsory, we recommend that you continue to run the SMP step to add the JCL that you have used to assemble modules to SMP as JCLIN (see Figure 135). This is useful so that if maintenance is later applied that affects the modules suffixed by the DBCTL region, they are then reassembled by SMP/E.

```
*  JCL in IMS.SJIMSD.DBDC.JCL($AJCLIN)

//AIGSMPE   PROC CSI=,PTS=,SYSOUT=
//SMPE      EXEC PGM=GIMSMP,REGION=18M,PARM='DATE=U'
//SMPCSI      DD DSN=&CSI,DISP=SHR
//SMPRPT      DD SYSOUT=&SYSOUT
//SMPOUT      DD SYSOUT=&SYSOUT
//SMPLIST     DD SYSOUT=&SYSOUT
//SMPPTS      DD DSN=&PTS,DISP=SHR
//SMPLOG      DD SYSOUT=&SYSOUT
//SMPHOLD     DD DUMMY
//SMPWRK1     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=3120,RECFM=FB,LRECL=80)
//SMPWRK2     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=3120,RECFM=FB,LRECL=80)
//SMPWRK3     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=3120,RECFM=FB,LRECL=80)
//SMPWRK4     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=3120,RECFM=FB,LRECL=80)
//SMPWRK5     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=6144,RECFM=U,DSORG=PO)
//SMPWRK6     DD UNIT=SYSDA,SPACE=(CYL,(10,10,200)),
//            DCB=(BLKSIZE=6144,RECFM=U,DSORG=PO)
//*
//SYSPRINT    DD SYSOUT=&SYSOUT
//SYSUT1      DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT2      DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT3      DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT4      DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//          PEND
//*
//*  JCLIN FOR IMSGEN STAGE 2
//*
//JCLIN     EXEC AIGSMPE,CSI='IMS610.GLOBAL.CSI',
//              PTS='IMS610.SMPPTS',
//              SYSOUT='X'
//SMPJCLIN   DD DSN=IMS.SJIMSD.STAGE1.SOURCE(STAGE2),
//              DISP=SHR,DCB=BUFNO=20
//SMPCNTL    DD  *
   SET BDY(IMSTZ61).
   JCLIN ASM(PGM=ASMA90).
//
```

Figure 135. Sample JCLIN Job after IMSGEN Stage 2

B.8.3.2 Run the Security Generation

Continue to run the DBCTL security facility, called *security gen*. The security gen defines the security of IMS databases. It must be run even if security is initially switched off. Figure 136 on page 258 shows the sample JCL job.

B.8.3.3 Run the Offline Copy

Once the security generation has been completed, the new generation must be copied from the offline staging libraries to the online libraries. The job shown in Figure 137 achieves this.

```
*  JCL placed in IMS.SJIMSD.DBDC.JCL($7OFLCHN)

//IMSDGOFC JOB (B3562,W941),'IMS TE-DBDC',
//          MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//  JCLLIB ORDER=IMS.SJIMSD.PROCLIB
//**************************************************************
//**USES ONLINE CHANGE UTILITY TO CHANGE BOTH SETS OF MODBLKS/MATRIX
//**     OFFLINE WHILE IMS IS DOWN.
//**************************************************************
//*
//OLCUTL PROC TYPE=,IN=,OUT=,SOUT='*',
//       SYS=,SYS2=SJIMSD
//*
//S      EXEC PGM=DFSUOCU0,PARM=(&TYPE,&IN,&OUT)
//STEPLIB  DD DISP=SHR,DSN=IMS.&SYS2..RESLIB
//MODBLKS  DD DSN=IMS.&SYS2..MODBLKS,DISP=SHR
//MODBLKSA DD DSN=IMS.&SYS2..MODBLKSA,DISP=SHR
//MODBLKSB DD DSN=IMS.&SYS2..MODBLKSB,DISP=SHR
//IMSACB   DD DSN=IMS.&SYS2..&SYS.ACBLIB,DISP=SHR
//IMSACBA  DD DSN=IMS.&SYS2..&SYS.ACBLIBA,DISP=SHR
//IMSACBB  DD DSN=IMS.&SYS2..&SYS.ACBLIBB,DISP=SHR
//MATRIX   DD DSN=IMS.&SYS2..MATRIX,DISP=SHR
//MATRIXA  DD DSN=IMS.&SYS2..MATRIXA,DISP=SHR
//MATRIXB  DD DSN=IMS.&SYS2..MATRIXB,DISP=SHR
//MODSTAT  DD DSN=IMS.&SYS2..MODSTAT,DISP=SHR
//SYSPRINT DD SYSOUT=&SOUT
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//COPYCTL  DD DSN=&&COPYCTL,DISP=(NEW,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//       PEND
//**********************************************************************
//* IMS ONLINE CHANGE UTILITY
//**********************************************************************
//OLC1A  EXEC OLCUTL,IN=S,OUT=A,TYPE=MATRIX
//OLC1B  EXEC OLCUTL,IN=S,OUT=B,TYPE=MATRIX
//OLC2A  EXEC OLCUTL,IN=S,OUT=A,TYPE=MODBLKS
//OLC2B  EXEC OLCUTL,IN=S,OUT=B,TYPE=MODBLKS
//
```

Figure 137. Sample JCL for Library Copy

```
*  JCL placed in IMS.SJIMSD.DBDC.JCL($4SECGEN)

//IMSDGSEC JOB (B3562,W941),'IMS TE-DBDC',
//         MSGLEVEL=(1,1),MSGCLASS=T,CLASS=A
//AIGSEC   PROC RESLIB=,MODBLKS=,MATRIX=,
//           RSUF=
//*
//* IMS Security gen proper
//*
//S       EXEC PGM=DFSISMP0,PARM='UPDATE,&RSUF',REGION=4M
//STEPLIB   DD DISP=SHR,DSN=&RESLIB
//          DD DISP=SHR,DSN=&MODBLKS
//SYSPRINT  DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=6144,LRECL=125)
//SYSPUNCH  DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,BUFNO=40)
//SYSLIN    DD UNIT=SYSDA,SPACE=(TRK,(5,1)),DISP=(,PASS),
//          DCB=(RECFM=FB,BLKSIZE=0,LRECL=80)
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,2)),
//          BLKSIZE=0,RECFM=FB,LRECL=100
//SYSUT2    DD UNIT=(SYSDA,SEP=SYSUT1),SPACE=(CYL,(5,2)),DCB=*.S.SYSUT1
//SYSIN     DD DISP=(SHR,PASS),BUFNO=40,DSN=IMS.SECURITY
//*
//* Assemble the Security matrix
//*
//      IF RC <= 4 THEN
//C       EXEC PGM=ASMA90,PARM='OBJ,NODECK',REGION=4M
//SYSPRINT  DD DUMMY,DCB=(RECFM=FB,LRECL=121,BLKSIZE=0)
//SYSLIN    DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,BUFNO=40)
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSIN     DD DSN=*.S.SYSPUNCH,DISP=(OLD,DELETE),BUFNO=40
//*
//* Link the Security matrix
//*
//L        EXEC PGM=IEWL,PARM='LIST,NE,OL',REGION=4M
//SYSPRINT  DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=0)
//SYSLMOD   DD DISP=OLD,DSN=&MATRIX
//INPUT     DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSLIN    DD DSN=*.S.SYSLIN,DISP=(OLD,DELETE)
//      ENDIF
//   PEND
//*****************************************************
//SECURITY  EXEC AIGSEC,
//            RESLIB='IMS.SJIMSD.RESLIB',
//            MATRIX='IMS.SJIMSD.MATRIX',
//            MODBLKS='IMS.SJIMSD.MODBLKS',
//            RSUF=D
//S.SYSIN  DD *
)( AGN BMP01
   AGPSB ALL
/*
```

Figure 136. Sample JCL for Assembling Security Matrix for DBCTL

B.9 Creating ACBLIB for IMS DBCTL

IMS DBCTL region uses the ACB library instead of the PSB and DBD libraries, as on the VSE local DLI environment. ACB thus merges information from both the PSB and DBD. These are the standard steps to create the ACB library:

1. Use standard DBDGEN procedure to create DBD members.
2. Use standard PSBGEN procedure to create PSB members.
3. Use IMS ACB maintenance utility (ACBGEN) to build the ACB library.
4. Copy the result to the online staging library.

All generation procedures are created by the IMS full generation process and are stored on the IMS procedure library along with others. *IMS/ESA, Installation Volume I, GC26-8736*, gives detailed information about these procedures. Figure 138 shows a sample ACBGEN procedure for reference. The JCL shown builds ACBs for all PSBs in IMS.SDIMSD.PSBLIB. It also can be used to build a specific PSB. To copy the newly built ACB members to an online staging library, use the procedure shown on Figure 137 on page 257

```
* JCL placed in IMS.SJIMSD.DBDC.JCL(ACBINIT)

//ACBGEN PROC SOUT='*',COMP=,SYS1='SJIMSD.'
//G      EXEC PGM=DFSRRC00,PARM='UPB,&COMP'
//STEPLIB  DD DISP=SHR,DSN=IMS.&SYS1.RESLIB
//SYSPRINT DD SYSOUT=&SOUT
//IMS      DD DSN=IMS.&SYS1.PSBLIB,DISP=SHR
//         DD DSN=IMS.&SYS1.DBDLIB,DISP=SHR
//IMSACB   DD DSN=IMS.&SYS1.ACBLIB,DISP=OLD
//SYSUT3   DD UNIT=SYSDA,SPACE=(80,(100,100))
//SYSUT4   DD UNIT=SYSDA,SPACE=(256,(100,100)),
//         DCB=KEYLEN=8
//COMPCTL  DD DISP=SHR,
//         DSN=IMS.&SYS1.PROCLIB(DFSACBCP)
//    PEND
//    EXEC ACBGEN,COMP=POSTCOMP
//SYSIN DD *
    BUILD PSB=ALL
```

Figure 138. Sample JCL for ACBGEN

B.10 Frequently Used Procedures

We have included a few of the most frequently used procedures for reference. All procedures are created during IMS full-generation processing and are stored in the PROCLIB library. However, customization may be needed for your local environment.

B.10.1 COBOL Programming Procedure

Figure 139 shows a procedure to translate, compile, and link a COBOL application program. The comment within the procedure describes its usage well.

```
//DFHEITVL PROC SUFFIX=1$,          Suffix for translator module
//*
//* This procedure has been changed since CICS/ESA Version 3
//* Parameter INDEX2 has been removed
//*
//        INDEX='CICSTS12',          Qualifier(s) for CICS libraries
//        PROGLIB=&INDEX..SDFHLOAD,  Name of output library
//        DSCTLIB=&INDEX..SDFHCOB,   Name of private macro/DSECT lib
//        COMPHLQ='SYS1',            Qualifier(s) for COBOL compiler
//        OUTC=A,                    Class for print output
//        REG=4M,                    Region size for all steps
//        LNKPARM='LIST,XREF',       Link edit parameters
//        STUB='DFHEILIC',           Link edit INCLUDE for DFHECI
//        LIB='SDFHCOB',             Library
//        WORK=SYSDA                 Unit for work datasets
//*
//*       This procedure contains 4 steps
//*       1.   Exec the COBOL translator
//*            (using the supplied suffix 1$)
//*       2.   Exec the vs COBOL II compiler
//*       3.   Reblock &LIB(&STUB) for use by the link-edit step
//*       4.   Link-edit the output into dataset &PROGLIB
//*
//*       The following JCL should be used to execute this procedure
//*
//*       //APPLPROG EXEC DFHEITVL
//*       //TRN.SYSIN  DD *
//*          .
//*          . Application program
//*          .
//*       /*
//*       //LKED.SYSIN DD *
//*          NAME anyname(R)
//*       /*
//*
//*       Where   anyname   is the name of your application program.
//*       (Refer to the system definition guide for full details,
//*       including what to do if your program contains calls to
//*       the common programming interface.)
//*
```

Figure 139 (Part 1 of 2). Procedure for Compiling CICS COBOL Program

```
//TRN    EXEC PGM=DFHECP&SUFFIX,PARM='COBOL2',
//         REGION=&REG
//STEPLIB  DD DSN=&INDEX..SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=&OUTC
//SYSPUNCH DD DSN=&&SYSCIN,DISP=(,PASS),UNIT=&WORK,
//         DCB=BLKSIZE=400,SPACE=(400,(400,100))
//*
//COB    EXEC PGM=IGYCRCTL,REGION=&REG,
//       PARM='NODYNAM,LIB,OBJECT,RENT,RES,APOST,MAP,XREF'
//STEPLIB  DD DSN=&COMPHLQ..COB2COMP,DISP=SHR
//SYSLIB   DD DSN=&DSCTLIB,DISP=SHR
//         DD DSN=&INDEX..SDFHCOB,DISP=SHR
//         DD DSN=&INDEX..SDFHMAC,DISP=SHR
//         DD DSN=&INDEX..SDFHSAMP,DISP=SHR
//SYSPRINT DD SYSOUT=&OUTC
//SYSIN    DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),
//         UNIT=&WORK,SPACE=(80,(250,100))
//SYSUT1   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT2   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT3   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT4   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT5   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT6   DD UNIT=&WORK,SPACE=(460,(350,100))
//SYSUT7   DD UNIT=&WORK,SPACE=(460,(350,100))
//*
//COPYLINK EXEC PGM=IEBGENER,COND=(7,LT,COB)
//SYSUT1   DD DSN=&INDEX..&LIB(&STUB),DISP=SHR
//SYSUT2   DD DSN=&&COPYLINK,DISP=(NEW,PASS),
//         DCB=(LRECL=80,BLKSIZE=400,RECFM=FB),
//         UNIT=&WORK,SPACE=(400,(20,20))
//SYSPRINT DD SYSOUT=&OUTC
//SYSIN    DD DUMMY
//*
//LKED   EXEC PGM=IEWL,REGION=&REG,
//            PARM='&LNKPARM',COND=(5,LT,COB)
//SYSLIB   DD DSN=&INDEX..SDFHLOAD,DISP=SHR
//         DD DSN=&COMPHLQ..COB2CICS,DISP=SHR
//         DD DSN=&COMPHLQ..COB2LIB,DISP=SHR
//SYSLMOD  DD DSN=&PROGLIB,DISP=SHR
//SYSUT1   DD UNIT=&WORK,DCB=BLKSIZE=1024,
//         SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=&OUTC
//SYSLIN   DD DSN=&&COPYLINK,DISP=(OLD,DELETE)
//         DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD DDNAME=SYSIN
//
```

Figure 139 (Part 2 of 2). Procedure for Compiling CICS COBOL Program

B.10.2 DLI Batch Processing Procedure

Figure 140 shows a procedure to run a DL/I batch job.

```
//       PROC MBR=TEMPNAME,PSB=,BUF=7,
//            SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,
//            SRCH=0,CKPTID=,MON=N,LOGA=0,FMTO=T,
//            IMSID=,SWAP=,DBRC=,IRLM=,IRLMNM=,
//            BKO=N,IOB=,SSM=,APARM=,
//            RGN=2048K,
//            SOUT=A,LOGT=2400,SYS2=,
//            LOCKMAX=,GSGNAME=,TMINAME=
//*
//G      EXEC PGM=DFSRRC00,REGION=&RGN,
//            PARM=(DLI,&MBR,&PSB,&BUF,
//            &SPIE&TEST&EXCPVR&RST,&PRLD,
//            &SRCH,&CKPTID,&MON,&LOGA,&FMTO,
//            &IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM,
//            &BKO,&IOB,&SSM,'&APARM',
//            &LOCKMAX,&GSGNAME,&TMINAME)
//STEPLIB  DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//         DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
//DFSRESLB DD DSN=IMS.&SYS2.RESLIB,DISP=SHR
//IMS      DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
//         DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//IEFRDER  DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),
//         UNIT=(&LOGT,,DEFER),
//         DCB=(RECFM=VB,BLKSIZE=1920,
//         LRECL=1916,BUFNO=2)
//IEFRDER2 DD DSN=IMSLOG2,DISP=(,KEEP),VOL=(,,,99),
//         UNIT=(&LOGT,,DEFER,SEP=IEFRDER),
//         DCB=(RECFM=VB,BLKSIZE=1920,
//         LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&SOUT,
//         DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//         SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
```

Figure 140. Sample DLI Batch Procedure

B.11 Model JCLs

IMS stores a set of predefined JCLs, called *model jcl* in an online library. Typical model JCL contains many symbols that can be manipulated by the system. For example, when the system log needs archiving or a database requires an image copy, the system uses these model JCLs to create the real JCL and submit jobs internally to the system to execute the function accordingly.

Listings in this section are generated by the IMS full-generation process and use standard product utility programs. Many of these need to be customized to suit your local environment. Some can even be replaced by another product, such as IBM DB Tools.

B.11.1 Log Archive Model JCL

This model JCL, Figure 141 on page 264, is used when DBCTL region log data sets are full and require archive processing. Further customization may be required for your local environment, such as single or dual logging, tape or OS/390 SMS manage devices, and so on.

```
//ARC%SSID JOB (B3562,W941),'ARCHIVE %SSID',
//         CLASS=3,MSGCLASS=A,MSGLEVEL=(1,1),
//         TIME=(1440),REGION=6M
//*
//*                 JCL FOR ARCHIVE UTILITY
//*                 -----------------------
//*
//*  JOB GENERATED BY DBRC (STARTED TASK OR BATCH UTILITY) FROM
//*  SKELETON JCL IN IMSVS.PROCLIB(ARCHJCL).
//*
//*  RECONS DYNAMICALLY ALLOCATED FROM MDS MEMBERS IN IMS.*.USERLIB
//*
//*  AMENDMENT RECORD
//* ================================================================
//* 07/03/97  NM   CREATED FROM JCL ON TSODA
//*
//*-------------------------------------------------------------------
//* STEP: ARCHIVE - RUN IMS UTILITY TO ARCHIVE OLDS TO SLDS
//*-------------------------------------------------------------------
//*
//* SET MIDDLE LEVEL QUALIFIER FOR DATASET NAMES.
//*
//      SET SYS1=SJIMSD
//ARCHIVE  EXEC PGM=DFSUARCO,PARM='%SSID,DBRC=YES'
//STEPLIB   DD DISP=SHR,DSN=IMS.&SYS1..USERLIB   DYNAL FOR RECONS
//          DD DISP=SHR,DSN=IMS.&SYS1..RESLIB    NEED A 6.1 RESLIB
//SYSPRINT  DD DISP=(NEW,PASS),DSN=&&LIST,
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6118)
//SYSUDUMP  DD SYSOUT=*
//*
%SELECT  OLDS(%SSID,(%DDNAMES))
//%OLDSDDN  DD DSN=%OLDSDSN,DISP=SHR,DCB=BUFNO=40
%ENDSEL
//*
//DFSSLOGP  DD DSN=IMS.&SYS1..SLDSP.D%ARDATE.T%ARTIME.V%ARVERS,
//          SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA,DISP=(,CATLG),
//          DCB=(IMS.&SYS1..OLPOO,LRECL=22520,BLKSIZE=30720,RECFM=VBS)
//SYSIN     DD *
SLDS FEOV(08000)
//*-------------------------------------------------------------------
//* PRINT ARCHIVE LISTING IF THE PREVIOUS STEP DIDN'T
//*-------------------------------------------------------------------
//PRINT    EXEC PGM=IEBGENER
//SYSUT1    DD DISP=(SHR,PASS),DSN=&&LIST
//SYSUT2    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//*-------------------------------------------------------------------
//* ALWAYS DELETE THE LISTING DATASET
//*---------------------------------------------
//DELETE   EXEC PGM=IEFBR14
//LISTIN    DD DISP=(OLD,DELETE),DSN=&&LIST
```

Figure 141. ARCHJCL Skeleton JCL to Archive OLDS Data Set

B.11.2 Database Image Copy Model JCL

Figure 142 shows model JCL that is managed by IMS DBRC GENJCL.IC processing to perform database image copy functions. Notice that the model JCL uses the standard IMS image-copy utility program to perform the task. You may replace this with another product, such as IBM DB Tools, to perform the same task.

```
//*
//*                JCL FOR IMAGE COPY UTILITY
//*                --------------------------
//*
//*  JOB GENERATED BY DBRC (STARTED TASK OR BATCH UTILITY) FROM
//*  SKELETON JCL IN IMSVS.PROCLIB(ARCHJCL).
//*
//*  RECONS DYNAMICALLY ALLOCATED FROM MDS MEMBERS IN IMS.*.USERLIB
//*
//*
//      SET SYS1='SJIMSD'
%DELETE (%IVPCIC NE '')
//IC%STPNO  EXEC PGM=DFSUDMP0,PARM='DBRC=Y'
%ENDDEL
%DELETE (%IVPCIC EQ '')
//IC%STPNO  EXEC PGM=DFSUDMP0,PARM='DBRC=Y,CIC'
%ENDDEL
//STEPLIB   DD DISP=SHR,DSN=IMS.&SYS1..USERLIB
//          DD DISP=SHR,DSN=IMS.&SYS1..RESLIB
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//IMS       DD DSN=IMS.&SYS1..DBDLIB,DISP=SHR
%SELECT  DBDS((%DBNAME,%DBDDN))
%DELETE  (%DBADSAV NE 'AVAIL')
//%DBADDN   DD DSN=%DBDSN,DISP=SHR
%ENDDEL
%DELETE  (%DBADSAV NE '')
//%DBDDN    DD DSN=%DBDSN,DISP=SHR
%ENDDEL
%ENDSEL
//DATAOUT1  DD DISP=OLD,DSN=%ICDSN1
%DELETE  (%COPIES EQ '1')
//DATAOUT2  DD DISP=OLD,DSN=%ICDSN2
%ENDDEL
//SYSIN     DD *
%ICSYSIN
/*
```

Figure 142. ICJCL — Skeleton JCL to Run an Image Copy of a Database

B.11.3 Database Recovery Model JCL

Figure 143 shows model JCL that is managed by IMS DBRC GENJCL.RECOV processing to perform database recovery function. For the complete IMS DBRC function, see *IMS/ESA, DBRC Guide and Reference, SC26-8733* for detailed information.

```
//*
//*              JCL FOR RECOVERY.
//*             ===================
//*
//    SET SYS1=SJIMSD
//RCV%STPNO EXEC PGM=DFSRRC00,
//          PARM='UDR,DFSURDB0,%DBNAME,,,,,,,,,,Y'
//STEPLIB   DD DISP=SHR,DSN=IMS.&SYS1..USERLIB
//          DD DISP=SHR,DSN=IMS.&SYS1..RESLIB
//SYSPRINT  DD SYSOUT=*
//IMS       DD DISP=SHR,DSN=IMS.&SYS1..DBDLIB
//%DBDDN    DD DSN=%DBDSN,
%DELETE  (%DBDSAM EQ 'VSAM')
//             DISP=OLD,
//             DCB=BUFNO=10
%ENDDEL
%DELETE  (%DBDSAM NE 'VSAM')
//             DISP=OLD
%ENDDEL
%DELETE  (%ICDSN EQ '')
//DFSUDUMP  DD DSN=%ICDSN,
//             DISP=OLD,DCB=BUFNO=10
%ENDDEL
%DELETE  (%ICDSN NE '')
//DFSUDUMP  DD DUMMY
%ENDDEL
%DELETE  (%HICDSN EQ '')
//DFSVDUMP  DD DSN=%HICDSN,
//             DISP=OLD
%ENDDEL
%DELETE  (%HICDSN NE '')
//DFSVDUMP  DD DUMMY
%ENDDEL
%DELETE  (%PICDSN EQ '')
%SELECT IC((%DBNAME,%DBDDNM),(FROM(%RICRNTM),TO(%RCVTOTM)),PARTIAL)
//DFSICLG%CNTR DD DSN=%ICDSN,
//             DISP=OLD
%ENDSEL
%ENDDEL
%DELETE  (%PICDSN NE '')
//DFSICLG1  DD DUMMY
%ENDDEL
%DELETE  (%CADSN EQ '')
//DFSUCUM   DD DSN=%CADSN,
//             DISP=OLD,DCB=BUFNO=10
%ENDDEL
%DELETE  (%CADSN NE '')
//DFSUCUM   DD DUMMY
%ENDDEL
```

Figure 143 (Part 1 of 2). Model JCL for Recovery Database from Image Copy

```
%SELECT  RLDS((%DBNAME,%DBDDN),FROM(%DSLLGTM))
//DFSULOG   DD DSN=%LOGDSN,
//             DCB=RECFM=VB,
//             DISP=OLD
%ENDSEL
%DELETE  (%LOGSEL EQ 'YES')
//DFSULOG   DD DUMMY
%ENDDEL
%DELETE  (%TRACK EQ 'NO')
//DFSTRCV   DD DSN=??????,
//             DISP=OLD
%ENDDEL
//DFSVSAMP  DD DISP=SHR,
//             DSN=IMS.&SYS1..PROCLIB(DFSVSMDB)
//SYSIN     DD *
%RCSYSIN
%DELETE  (%HICDSN EQ '')
%RICRNTMI              %HICDSN
%ENDDEL
%DELETE  (%PICDSN EQ '')
%SELECT IC((%DBNAME,%DBDDNM),(FROM(%RICRNTM),TO(%RCVTOTM)),PARTIAL)
%ICTIME               %ICDSN
%ENDSEL
%ENDDEL
/*
```

Figure 143 (Part 2 of 2). Model JCL for Recovery Database from Image Copy

B.12 Manual Operation

Figure 144 and Figure 145 on page 269 show manual image copy and recovery JCLs for reference. Manual operation is not recommended in general; this is for emergency use only.

```
//IMSDDB   JOB (999,POK),'DBRC IMSD',
//         CLASS=A,MSGCLASS=T,MSGLEVEL=(1,1),
//         TIME=(1440),REGION=6M
//*
//*
//*
//*                 JCL FOR IMAGE COPY UTILITY
//*                 --------------------------
//*
//*  JOB GENERATED BY DBRC (STARTED TASK OR BATCH UTILITY) FROM
//*  SKELETON JCL IN IMSVS.PROCLIB(ARCHJCL).
//*
//*  RECONS DYNAMICALLY ALLOCATED FROM MDS MEMBERS IN IMS.*.USERLIB
//*
//*
//       SET SYS1='SJIMSD'
//IC1  EXEC PGM=DFSUDMP0,PARM='DBRC=Y'
//STEPLIB   DD DISP=SHR,DSN=IMS.&SYS1..USERLIB
//          DD DISP=SHR,DSN=IMS.&SYS1..RESLIB
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//IMS       DD DSN=IMS.&SYS1..DBDLIB,DISP=SHR
//CZFBTBE    DD DSN=CASE.ZJLS.CZFFTBE,DISP=SHR
//DATAOUT1  DD DISP=OLD,DSN=CASE.ZJLS.CZFFTBE.IC102
//SYSIN     DD *
D1 CZFBTBE  CZFBTBE  DATAOUT1
/*
```

Figure 144. Sample JCL for Manual Image Copy

```
//RESTRDB JOB (B3562,W941),'IMS TE-DBDC',
//        MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//*
//RECOVER EXEC PGM=DFSRRC00,PARM='UDR,DFSURDB0,DBD1'
//STEPLIB   DD DISP=SHR,DSN=IMS.SJIMSD.USERLIB
//          DD DISP=SHR,DSN=IMS.SJIMSD.RESLIB
//IMS       DD DISP=SHR,DSN=IMS.SJIMSD.DBDLIB
//SYSPRINT  DD SYSOUT=*
//DFSUINPT  DD DISP=OLD,DSN=IMS.DB.FILE1.EXP,
//          UNIT=TAPE,VOL=SER=TAPDIC,LABEL=(1,SL)
//DFSUCUM0  DD DUMMY
//DFSULOG   DD DSN=IMS.SJIMSD.SLDSP.D*.T*.V*  <===================
//          DD DSN=IMS.SJIMSD.SLDSP.D*.T*.V*  as many as necessary
//          DD DSN=IMS.SJIMSD.SLDSP.D*.T*.V*  <===================
//DBD1      DD DISP=SHR,DSN=USER1.DB.FILE1
//SYSIN     DD *
NOSEQCK
S  DBD1     DBD1     DFSUINPT
/*
//
```

Figure 145. Manual Recovery of Database from Image Copy and Logs

To demonstrate an alternative, Figure 146 shows a database backup job using the IBM DB Tool instead of the IMS product utility program. For detailed information, refer to the DB Tools product publication and ITSO redbooks (in preparation) *IMS DB Tools Vol I, SG24-5166*, and *IMS DB Tools Vol II, SG24-5242*.

```
//BACKUPDB JOB (B3562,W941),'IMS TE-DBDC',
//          MSGLEVEL=(1,1),MSGCLASS=T,TIME=(,30)
//FABHULU PROC IMSDSN=IMSVS,DBTLMD='DBT.SFABLMD0',SOUT='*',
//             MBR=TEMPNAME,CTRL=DFSRRC00,DBD=,BUF=7,
//             SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,CKPTID=,
//             MON=N,LOGA=0,FMTO=T,IMSID=,SWAP=,DBRC=,IRLM=,IRLMNM=
//*
//G       EXEC PGM=FABHX034,PARM=('&CTRL/ULU',&MBR,&DBD,&BUF,
//             &SPIE&TEST&EXCPVR&RST,&PRLD,&SRCH,&CKPTID,&MON,
//             &LOGA,&FMTO,&IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM)
//STEPLIB   DD DSN=&IMSDSN..USERLIB,DISP=SHR
//          DD DSN=&IMSDSN..RESLIB,DISP=SHR
//          DD DSN=&DBTLMD,DISP=SHR
//DFSRESLB  DD DSN=&IMSDSN..RESLIB,DISP=SHR
//IMS       DD DSN=&IMSDSN..DBDLIB,DISP=SHR
//PROCLIB   DD DSN=&IMSDSN..PROCLIB,DISP=SHR
//IEFRDER   DD DSN=NULLFILE,
//          DISP=(,KEEP),VOL=(,,,99),UNIT=(TAPE,,DEFER),
//          DCB=(RECFM=VBS,LRECL=3964,BLKSIZE=3968,DEN=3)
//SYSUDUMP  DD SYSOUT=&SOUT
//IMSMON    DD DUMMY
//HSSRSTAT  DD SYSOUT=&SOUT  REMOVE DD-STATEMENT IF NO STATS WANTED
//HSSRTRAC  DD SYSOUT=&SOUT
//HSSRSNAP  DD SYSOUT=&SOUT
//        PEND
//*
//        EXEC FABHULU,MBR=FABHURG1,DBD=DBD1,
//        IMSID=IMSD,DBRC=YES
//SYSPRINT  DD SYSOUT=*
//SYSUT2    DD DISP=(NEW,CATLG,DELETE),
//          DSN=IMS.USER.DB.FILE1.BACKUP,
//          UNIT=TAPE,LABEL=(,SL)
//DFSVSAMP  DD DISP=DHR,DSN=IMS.SJIMSD.PROCLIB(DFSVSMDB)
//DBD1      DD DISP=SHR,DSN=USER1.DB.FILE1  Database DD as in DBD.
```

Figure 146. Database Backup by IMS DB Tools

Appendix C. COBOL FILE Key Status Values

We want to save application programmers from having to look through various manuals to find file status code, which is one of most time consuming tasks during application conversion, development, or testing. Therefore, we list the most common status code in Table 27 and Table 28 on page 272. For the status codes less often used, refer to product publications

Table 27. Status Key Values —— QSAM Files

OS/VS COBOL	IBM COBOL for MVS	Description
	04	Wrong length record. Successful completion
	05	Optional file not present. Successful completion
	07	NO REWIND/REEL/UNIT/FOR REMOVAL specified for OPEN or CLOSE, but file not on a reel/unit medium. Successful completion
00	00	Successful completion
10	10	At END (no next logical record). Successful completion
30	30	Permanent error
34	34	Permanent error File boundary violation
90	90	Other errors with no further information
90	35	Required nonoptional file not present
90	37	Device-type conflict
90	39	Conflict of fixed file attributes; OPEN fails
90	96	No file identification (No DD statement for the file)
92	38	OPEN attempted for file closed with lock
92	41	OPEN attempted for a file in OPEN mode
92	42	CLOSE attempted for a file not in OPEN mode
92	43	REWRITE attempted when last I/O statement was not READ
92	44	Attempt to rewrite a sequential file record with a record of a different size
92	46	Sequential READ attempted with no valid next record
92	47	READ attempted when file not in OPEN INPUT or I-O mode
92	48	WRITE attempted when file not in OPEN OUTPUT, I-O, or EXTEND mode
92	49	DELETE or REWRITE attempted when file not in OPEN I-O mode
92	92	Logic error

Table 28. IBM COBOL VSAM File Status Values		
OS/VS COBOL	**IBM COBOL for MVS**	**Description**
	14	On sequential READ for relative file, size of relative record number too large for relative key
00	00	Successful completion
00	04	Wrong length record. Successful completion
00	05	Optional file not present. Successful completion
02	02	Duplicate key, and DUPLICATES specified. Successful completion
10	10	At END (no next logical record). Successful completion
21	21	Invalid key for a VSAM indexed or relative file; sequence error
22	22	Invalid key for a VSAM indexed or relative file; duplicate key and duplicates not allowed
23	23	Invalid key for a VSAM indexed or relative file; no record found
24	24	Invalid key for a VSAM indexed or relative file; attempt to write beyond file boundaries IBM COBOL for a WRITE to a relative file, size of relative record number too large for relative key
30	30	Permanent error
90	37	Attempt to open a file not on a mass storage device
90	90	Other errors with no further information
91	91	VSAM password failure
92	41	OPEN attempted for a file in OPEN mode
92	42	CLOSE attempted for a file not in OPEN mode
92	43	REWRITE attempted when last I/O statement was not READ or DELETE
92	47	READ attempted when file not in OPEN INPUT or I-O mode
92	48	WRITE attempted when file not in OPEN OUTPUT, I-O, or EXTEND mode
92	49	DELETE or REWRITE attempted when file not in OPEN I-O mode
93	93	VSAM resource not available
93 and 96	35	Required nonoptional file not present
94	46	Sequential READ attempted with no valid next record
95	39	Conflict of fixed file attributes; OPEN fails
95	95	Invalid or incomplete VSAM file information
96	96	No file identification (no DD statement for this VSAM file)
97	97	OPEN statement execution successful; file integrity verified

Appendix D. CICS Command-Level Equivalents of MACRO Calls

For those users who still have macro calls on their application programs, we present a summary of the most commonly used CICS application macro and their equivalent command-level function. Because of time and experience limitations, we may miss some macros less commonly used. Refer to product publications for complete listings and detailed usage information.

Table 29 (Page 1 of 3). CICS Macro Calls and Command-Level Equivalents

CICS Application Macro	Command-Level Equivalent
DFHBFTA	
DFHBIF	
TYPE=DEEDIT	BIF DEEDIT
DFHBMS	
TYPE=CHECK TYPE=IN TYPE=MAP TYPE=OUT TYPE=OUT,MAP= TYPE=PAGEBLD TYPE=PAGEOUT TYPE=PURGE TYPE=RETURN TYPE=ROUTE TYPE=STORE TYPE=TEXTBLD	HANDLE CONDITION RECEIVE MAP RECEIVE MAP FROM SEND TEXT SEND MAP SEND MAP ACCUM SEND PAGE PURGE MESSAGE SEND{MAP|TEXT} SET ROUTE SEND{MAP|TEXT} PAGING SEND TEXT ACCUM
DFHDC	
TYPE=CICS TYPE=COMPLETE TYPE=PARTIAL,LIST=PROGRAM TYPE=PARTIAL,LIST=TERMINAL TYPE=PARTIAL,LIST=TRANSACTION TYPE=PARTIAL,LIST=SEGMENT TYPE=TRANSACTION	DUMP TABLES DUMP COMPLETE DUMP PROGRAM DUMP TERMINAL DUMP STORAGE DUMP FROM DUMP TASK
DFHDI	
TYPE=ABORT TYPE=ADD TYPE=CHECK TYPE=END TYPE=ERASE TYPE=NOTE TYPE=QUERY TYPE=RECEIVE TYPE=REPLACE TYPE=SEND TYPE=WAIT	ISSUE ABORT ISSUE ADD HANDLE CONDITION ISSUE END ISSUE ERASE ISSUE NOTE ISSUE QUERY ISSUE RECEIVE ISSUE REPLACE ISSUE SEND ISSUE WAIT
DFHFC	
TYPE=CHECK TYPE=DELETE (DL/I types) TYPE=ESETL TYPE=GET	HANDLE CONDITION DELETE RIDFLD ENDBR READ

Table 29 (Page 2 of 3). CICS Macro Calls and Command-Level Equivalents	
CICS Application Macro	**Command-Level Equivalent**
TYPE=GET,TYPOPER=UPDATE	READ UPDATE
TYPE=GETAREA	
TYPE=GETNEXT	READNEXT
TYPE=GETPREV	READPREV
TYPE=PUT,TYPOPER=DELETE	DELETE
TYPE=PUT,TYPOPER=NEWREC	WRITE
TYPE=PUT,TYPOPER=UPDATE	REWRITE
TYPE=RELEASE	UNLOCK
TYPE=RESETL	RESETBR
TYPE=SETL	STARTBR
DFHIC	
TYPE=CANCEL	CANCEL
TYPE=CHECK	HANDLE CONDITION
TYPE=GET	RETRIEVE
TYPE=GETIME	ASKTIME
TYPE=INITIATE	START
TYPE=POST	POST
TYPE=PUT	START FROM
TYPE=RETRY	RETRIEVE
TYPE=WAIT	DELAY
DFHJC	
TYPE=CHECK	HANDLE CONDITION
TYPE=GETJCA	
TYPE=PUT	JOURNAL WAIT
TYPE=WAIT	WAIT JOURNAL
TYPE=WRITE	JOURNAL
DFHKC	
TYPE=ATTACH	
TYPE=CHAP	
TYPE=DEQ	DEQ
TYPE=ENQ	ENQ
TYPE=NOPURGE	
TYPE=PURGE	
TYPE=WAIT	SUSPEND
TYPE=WAIT,ECADDR	WAIT EVENT
DFHMDF	
DFHMDI	
DFHMSD	
DFHPC	
TYPE=ABEND	ABEND
TYPE=CHECK	HANDLE CONDITION
TYPE=COBADDR	
TYPE=DELETE	RELEASE
TYPE=LINK	LINK
TYPE=LOAD	LOAD
TYPE=RESETXIT	HANDLE ABEND RESET
TYPE=RETURN	RETURN
TYPE=SETXIT	HANDLE ABEND
TYPE=XCTL	XCTL
DFHSC	

Table 29 (Page 3 of 3). CICS Macro Calls and Command-Level Equivalents	
CICS Application Macro	**Command-Level Equivalent**
TYPE=FREEMAIN TYPE=GETMAIN	FREEMAIN GETMAIN
DFHSP	
TYPE=USER TYPE=ROLLBACK	SYNCPOINT SYNCPOINT ROLLBACK
DFHTC	
TYPE=CBUFF TYPE=CONVERSE TYPE=COPY TYPE=DISCONNECT TYPE=EODS TYPE=ERASEAUP TYPE=GET TYPE=PAGE TYPE=PASSBK TYPE=PRINT TYPE=PROGRAM TYPE=PUT TYPE=READ TYPE=READB TYPE=READL TYPE=RESET TYPE=SIGNAL TYPE=WAIT TYPE=WRITE TYPE=WRITEL	SEND CBUFF CONVERSE ISSUE COPY ISSUE DISCONNECT ISSUE EODS ISSUE ERASEAUP RECEIVE SEND PASSBK ISSUE PRINT ISSUE LOAD SEND WAIT RECEIVE(WAIT assumed) RECEIVE BUFFER RECEIVE LEAVEKB ISSUE RESET WAIT SIGNAL WAIT TERMINAL SEND SEND LEAVEKB
DFHTD	
TYPE=CHECK TYPE=FEOV TYPE=GET TYPE=PURGE TYPE=PUT	HANDLE CONDITION READQ TD DELETEQ TD WRITEQ TD
DFHTR	
TYPE=ENTRY TYPE=OFF TYPE=ON	ENTER TRACE OFF TRACE ON
DFHTS	
TYPE=CHECK TYPE=GET TYPE=GETQ TYPE=PURGE TYPE=PUT TYPE=PUTQ TYPE=RELEASE	HANDLE CONDITION READQ TS READQ TS DELETEQ TS WRITEQ TS WRITEQ TS DELETEQ TS
DFHWTO	
TYPE=0	WRITE OPERATOR

Appendix E. Security Parameters

IBM supplies RACF for use on OS/390 as a security management facility. It is normally installed as an integral part of the operating system.

In general, RACF is managed by a group responsible for the entire installation security. The RACF administrator defines *Users*, then collects them into groups. Everything else (transaction, file, connection, PSB, program, application, and so on) is resources, and has associated RACF profiles defined. Users and groups have to have permission defined in RACF to be able to use them.

E.1 IMS and CICS Security Parameters

RACF can return three possible values: allowed, not allowed, and don′t know. You decide when you code the CICS and IMS start-up parameters what resources you need to protect. There are several places your user can specify security parameters:

1. IMS Generation Processing.

   The macros are:

   - SECURITY macro. Specifies the resource name suffix and what will be protected. For DBCTL, users can choose only whether or not to protect AGNs; that is, which IMS BMPs need permit access to databases and PSBs. Initial values recommended are these:

     ```
     SECURITY RCLASS=IMS,TYPE=(RACFAGN)
     ```

     This says that our resource names in IMS will end in ′-IMS′ and that we will use RACF to check AGN security. TYPE=NOAGN would say that we will not make these checks.

   - IMSGEN and COMM macros. Neither of these have relevant options unless user are also running IMS TM.

2. The IMS Security Generation. This is principally concerned with command security in DBCTL.

3. IMS Start-up Parameters. These are kept in a member of DFSPBxxx in an online PROCLIB library. In our case, the library is:

   ```
   IMS.SJIMSD.PROCLIB(DFSPBSJD)
   ```

   The relevant ones are:

   - AOIS. Determines whether RACF is to be used for operator command security. Set AOIS=N, R, os S. N means that programs cannot issue ICMD calls; R means that RACF decides; and S means no checking. This was initially set to AOIS=S.

   - ISIS. Set to say whether you want resource checking. Values are ISIS=0, or 1. 0 means no checking, 1 means use RACF. This overrides the value in the IMS generation.

4. CICS Start-up Parameters. These are kept in members of an on-line data set point by the PARM field in the CICS region startup JCL. In our case, the file is:

```
CICS.SJCICSD.PARMLIB(OVERRIDES)
```

The relevant parameters are:

SEC Whether there is to be resource checking by RACF (SEC=YES) or not (SEC=NO). Initial value was set at NO. This means no security.

PLTPISEC Whether the RACF checking is to occur during PLT processing or not. Initially set to NO.

SECPRFX Whether the resource names are to be prefixed by the CICS name (SECPRFX) or not. The initial value is the default of NO. If you set it to YES, then when you define a resource to RACF it is unique to that CICS region. If you do set SECPRFX=NO, then a RACF definition for a resource (such as a transaction) will be effective for that resource in all CICS streams with SECPRFX=NO. So if you can access CEMT in one CICS region, you can access it anywhere.

XAPPC Whether to use RACF for APPC connection security. Set to YES or NO.

XCMD Whether to use RACF for command security. Set to YES or NO.

XDCT Whether to use RACF for transient data queue security. Set to YES or NO.

XFCT Whether to use RACF for file security. Set to YES or NO.

XJCT Whether to use RACF for journal security. Set to YES or NO.

XPCT Whether to use RACF to check whether your transaction has authority to start other transactions. Set to YES or NO.

XPPT Whether to use RACF to check whether your transaction has authority to LINK, LOAD, or XCTL other programs. Set to YES or NO.

XPSB Whether to use RACF for PSB security in CICS. Set to YES or NO.

XTRAN Whether to use RACF for transaction security. Set to YES or NO.

XTST Whether to use RACF for temporary storage security. Set to YES or NO.

XUSER Whether to use RACF for User security. Set to YES or NO.

5. CICS Resource Definitions.

   Transactions have three important parameters:

   - EXTSEC=NO. Allows you to exclude a transaction that would otherwise be subject to RACF security checking. For example, if your users have defined it to CICS but your RACF administrator has not defined it yet.
   - RESSEC=YES. This forces full resource checking for this transaction.
   - CMDSEC=YES. It invokes RACF to check SPI commands in program.

Terminals have two important parameters:

- USERID=. Users must specify this for printers. There is a default user, which this terminal is deemed to be signed on with. This replaces the old practice of assigning transaction classes to terminals such as consoles and printers. If you do not code this for a printer, the printer will not be able to sign on, and so will be unable to do any work.
- ATTACHSEC =LOCAL or =PERSISTENT or=VERIFY This defines how security checking is to be done for remote terminals. LOCAL means that you trust the link; PERSISTENT means that you verify a user′s ID yourself, but VERIFY means that the terminal must be signed on specifically to you system before you can allow the connection.

  A list of the various parameters relating to security for CICS resources is given in Table 30

E.2 RACF Resources for CICS and IMS DBCTL

We summarize here all the RACF resources that can be defined in the CICS TS and IMS DBCTL processing environment. Table 30 cross references the CICS resource to its related security parameters.

Table 30. CICS and DBCTL Resources Defined to RACF

CICS Resource	**Security Parameters**
FILE	
TRANSACTION	EXTSEC CMDSEC RESSEC
CONNECTION	SECURITYNAME ATTACHSEC BINDPASSSWORD BINDSECURITY USERID
TERMINAL	SECURITYNAME ATTACHSEC BINDPASSSWORD BINDSECURITY USERID

We show here the most common RACF resources for a CICS and DBCTL environment for reference. Assuming you set RCLASS=IMS in the IMSGEN SECURITY macro, and SECPRFX=NO in the CICS start-up parameters, then, Table 31 on page 280 shows all of the possible RACF resources that can be defined for this CICS and DBCTL system. There are also others, but those are not of interest to users migrating from VSE.

Table 31. Available RACF Resources for CICS and DBCTL

Resource Class	Resource Class Name
IMS resource classes with RCLASS=IMS	
Command resource class	CIMS
Command group resource class	DIMS
Database resource class	PIMS
Database group resource class	QIMS
Segment resource class	SIMS
Segment group resource class	UIMS
Field resource class	FIMS
Field group resource class	HIMS
Other resource class	OIMS
Other group resource class	WIMS
Application group name resource class	AIMS
Application resource class	APPL
CICS resource classes with SECPRFX=NO	
Transaction class	TCICSTRN
Transaction group class	GCICSTRN
PSB resource class	PCICSPSB
PSG group class	QCICSPSB
File resource class	FCICSFCT
File group class	HCICSFCT
Journal class	JCICSJCT
Journal group class	KCICSJCT
Transient Data class	DCICSDCT
Transient Data group class	ECICSDCT
Temporary Storage class	SCICSTST
Temporary Storage group	UCICSTST
Linked Program Class	MCICSPPT
Linked Program group	NCICSPPT
Started transaction class	ACICSPCT
Started transaction group	BCICSPCT
General RACF resource classes	
User	USER
File	DATASET

For further details see *CICS Transaction Server for OS/390 CICS RACF Security Guide, SC33-1701*, *IMS/ESA, Administration Guide: System, SC26-8730*, and *OS/390 RACF Command Reference, SC28-1919*.

Appendix F. Conversion Aid

F.1 COBOL Conversion Aid for VSE

COBOL and CICS/VS Command Level Conversion Aid (CCCA) for VSE (Program 5785-CCC) is an effective tool designed to make it easier to convert old COBOL source code and copy modules to the new COBOL standard. CCCA converts DOS/VS COBOL and COBOL 74 VS COBOL II (Release 3 and 4 (CMPR2)) source code to COBOL 85 Standard VS COBOL II Release 3 or 4 (NOCMPR2) or to IBM COBOL for VSE/ESA.

CCCA is designed to identify and convert incompatible source code, to reduce the effort required to convert programs, and to minimize conversion errors. In cases where a statement is no longer supported and has no equivalent statement in the target COBOL, CCCA flags the statement. The conversion process can be customized by users to meet unique conversion requirements. Installation and usage are easy, fast, and straightforward. CCCA key benefits are these:

- Identifies and converts source code.
- Reduces the effort required to convert programs.
- Minimizes conversion errors.
- Enhances programmer productivity during migration. CCCA provides facilities to:
  - Convert most syntax differences between OS/VS COBOL, DOS/VS COBOL, or VS COBOL II Release 1 or 2 and the current release of VS COBOL II and COBOL for MVS and VM and COBOL for VSE/ESA programs.
  - Convert EXEC CICS commands.
  - Remove or convert the base locator for linkage (BLL) section mechanism and references.
- Eliminates conflicts between user-defined names and words reserved for VS COBOL II.
- Converts both source programs and copy modules.
- Creates conversion management reports.
- Produces a statement-by-statement diagnostic listing showing the result of the conversion process for each program.
- Changes or creates COBOL conversion modules.
- Allows foreground conversion of CICS programs.
- Performs conversion from various levels of COBOL into other COBOL levels through an open converter design.
- Reads from PDSEs, not just PDSs.

F.2 CICS/VSE Application Migration Aid

IBM Hursley Transaction Systems Technical Support group in the United Kingdom provides application migration aid to assist users in converting CICS application from the macro-level API to the command-level API. It converts the simpler macros in an application program automatically. Also, it partially converts the more complex macros, issuing messages that guide users to complete the conversion. It runs as an offline batch utility on any VSE/ESA system.

F.2.1 The Processing Flows

Figure 147 on page 283 shows the conversion procedure.

The migration aid takes a macro-level source program as input and deals with the macros in the program in one of two ways:

- The aid converts the simpler macros in the program to equivalent command-level functions. The program is entirely converted by the migration aid utility.
- The aid cannot convert the complex macros; instead, it starts the conversion for each macro and issues a message prompting the user to supply missing data to complete the conversion.

The output is written to a data set, in most cases, with no assembler or compiler. The command-level program requires thorough testing before it can be put into production.

F.2.2 The Conversion Examples

The following examples show sample conversion:

- Simple conversion.

```
Source: DFHTD TYPE=PURGE,DESTID=CSML

Converted to COBOL output source:
        EXEC  CICS  DELETEQ  TO  QUEUE('CSML')
        END-EXEC

Converted  to PL/I output source:
        EXEC  CICS  DELETEQ  TO  QUEUE('CSML');
```

- Complex conversion that needs manual intervention:

```
Source: DFHTD TYPE=PUT,DESTID=CSML,TDADDR-TDOAVRL

Convert output:
        EXEC CICS WRITEQ TD QUEUE('CSML') FROM(????) LENGTH(???)
        *ERCMOP03 08 &mdasd. FROM() OPTION REQUIRED
        *ERCMOP04 08 &mdasd. LENGTH() VALUE REQUIRED
```

  Once the item is partially converted, users have to provide the required data to complete the conversion.

Users also need to remove system areas such as the common system area (CSA), terminal control table terminal entry (TCTTE), and so on, that are not supported by the command-level programming.

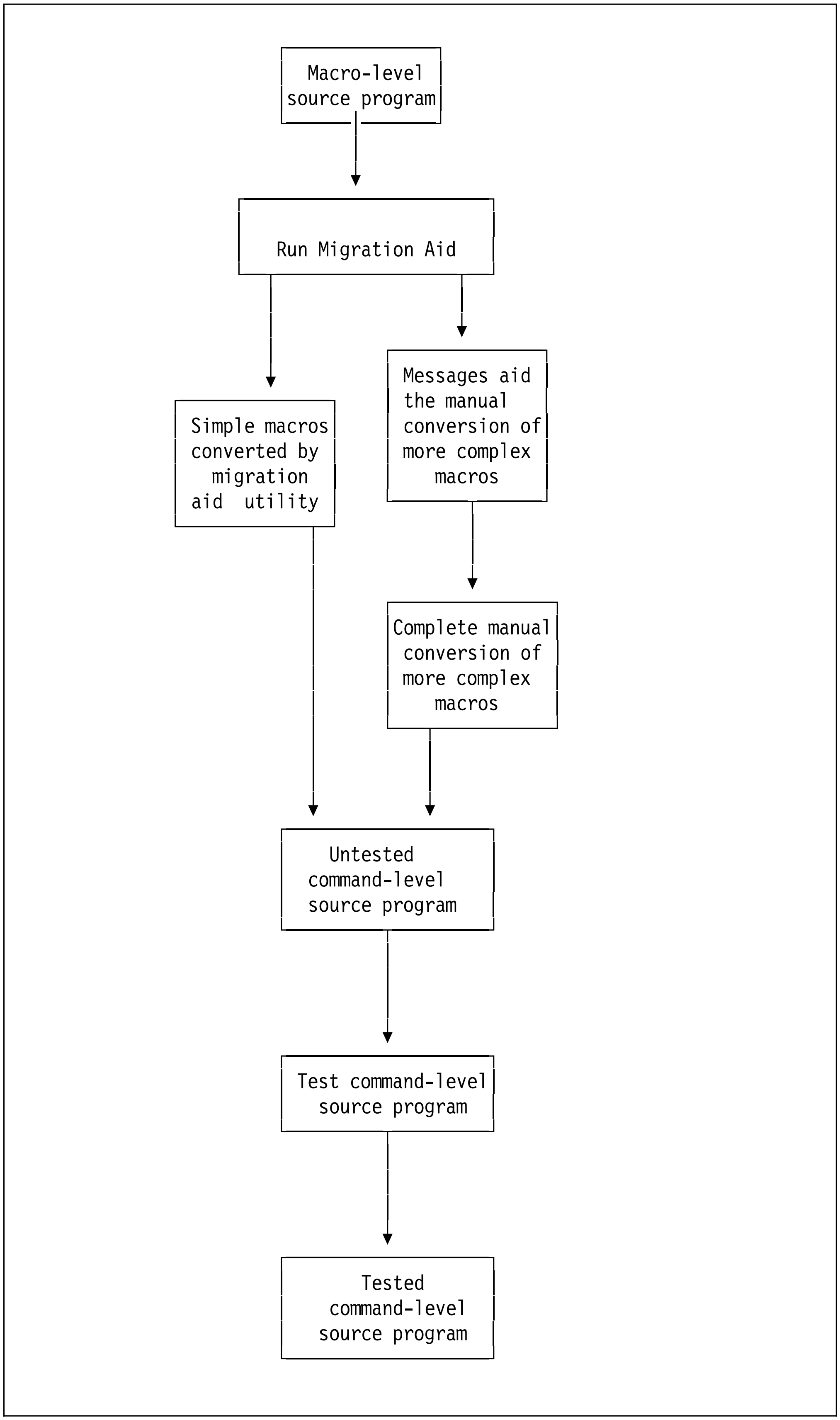


Figure 147. CICS Application Conversion Procedure

F.3 CICS Local DL/I to IMSGEN Conversion Aid

Dougie Lawson of the UK technical support group developed replacement macros for DFHDLDBD and DFHDLPSB. MVS local DL/I users will be able to convert their PPT and PCT macros source and produce the corresponding IMS Stage 1 source macros input for IMS generation processing. The replacement converts the local PSB and DBD only and displays the warning messages to ask users to delete the remote entries. It is very easy to use and no document is needed. Figure 148 on page 285 lists the source of the macros and Figure 149 on page 286 shows the sample job, input source, and converted output data.

```
DFHDLDBD Macro :-

         MACRO
&LABEL   DFHDLDBD  &INDEX,         POSITIONAL, FOR DOCUMENTATION ONLY  *
               &TYPE=,             TYPE OF ENTRY                       *
               &DBD=,              DBD NAME                            *
               &DLI=,                                                  *
               &DL1=,                                                  *
               &ACCESS=,           ACCESS LEVEL FOR DATASHARING        *
               &SUFFIX=            MODULE SUFFIX             @E8DOE @L1C
         AIF   ('&TYPE' NE 'ENTRY').NOTENT
         AIF   ('&ACCESS' EQ '').NOACESS
         PUNCH '         DATABASE DBD=&DBD,ACCESS=&ACCESS'
         AGO   .EXIT
.NOACESS ANOP
         PUNCH '         DATABASE DBD=&DBD'
         AGO   .EXIT
.NOTENT  ANOP
         AIF   ('&TYPE' NE 'FINAL').EXIT
         PUNCH '* Please delete the next two lines'
DFHIDMDO CSECT
&perios.EXIT    ANOP
         MEND


DFHDLPSB Macro :-

         MACRO
&LABEL   DFHDLPSB  &TYPE=,         TYPE OF ENTRY                       *
               &PSB=,              PSB NAME                            *
               &MXSSASZ=,          MAX SSA SIZE(REMOTE PSB)            *
               &RMTNAME=,          PSB NAME ON REMOTE SYS              *
               &SYSIDNT=,          REMOTE SYSTEM NAME                  *
               &DLI=,                                                  *
               &DL1=,                                                  *
               &SUFFIX=            MODULE SUFFIX                @BC5C11H
         AIF   ('&TYPE' NE 'ENTRY').NOTENT
         AIF   ('&SYSIDNT' NE '').REMOTE
         PUNCH '         APPLCTN PSB=&PSB,PGMTYPE=BATCH,SCHDTYP=PARALLE*
               L'
         AGO   .EXIT
.REMOTE  ANOP
         PUNCH '* REMOTE PSB &PSB - No IMS definition generated'
         AGO   .EXIT
.NOTENT  ANOP
         AIF   ('&TYPE' NE 'FINAL').EXIT
         PUNCH '* Please delete the next two lines'
DFHIDIRO CSECT
.EXIT    ANOP
         MEND
```

Figure 148. Local DL/I Conversion Macros

```
Execution JCL:

    //CHENST  JOB  (999,POK),CHENS,CLASS=A,NOTIFY=&SYSUID,
    //        MSGLEVEL=(1,1),MSGCLASS=X
    //*
    //       EXEC PGM=ASMA90,REGION=4M,PARM='NOOBJECT,DECK'
    //SYSPRINT DD SYSOUT=*
    //SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
    //SYSLIB   DD DSN=CHENS.IMS.MACROS,DISP=SHR
    //SYSPUNCH DD SYSOUT=*
    //SYSIN    DD DSN=CHENS.PDS.DATA(CICSDBD),DISP=SHR
------------------------------------------------------------

Input CICS/MVS local DL/I DBD macro source data:

     DFHDLDBD TYPE=INITIAL,DL1=3.1.0,SUFFIX=DL
     DFHDLDBD TYPE=ENTRY,DBD=XX1,ACCESS=RO
     DFHDLDBD TYPE=ENTRY,DBD=XX2,ACCESS=RD
     DFHDLDBD TYPE=ENTRY,DBD=XX3,ACCESS=UP
     DFHDLDBD TYPE=ENTRY,DBD=XX4,ACCESS=EX
     DFHDLDBD TYPE=ENTRY,DBD=XX5
     DFHDLDBD TYPE=FINAL
     END   DFHIDMDO
------------------------------------------------------------

Converted output IMS Stage 1 generation macro source data:


Assembly Start Time: 14.24.23 Stop Time: 14.24.23
Processor Time: 00.00.00.0038 Return Code 000

        DATABASE DBD=XX1,ACCESS=RO
        DATABASE DBD=XX2,ACCESS=RD
        DATABASE DBD=XX3,ACCESS=UP
        DATABASE DBD=XX4,ACCESS=EX
        DATABASE DBD=XX5
* Please delete the next two lines
```

Figure 149. Test Run on Conversion Macros

Appendix G. Available IMS Commands for DBCTL

DBCTL uses a subset of IMS commands for operation. The CICS-supplied transaction CDBM can act as an interface for most of the IMS DBCTL commands and acts as the DBCTL master console. For easy reference, we summarize here all possible DBCTL commands and the CICS--supplied equivalent transaction command in Table 32. Table 33 on page 289 also summarizes the most commonly used keywords of the DBCTL commands.

Table 32 (Page 1 of 2). DBCTL Operator Commands and CICS Equivalents

DBCTL Operator Command	CICS Equivalent	Valid with CDBM
/CHANGE	None	Yes
/CHECKPOINT (simple form)	ACTIVITY KEYPOINT	Yes
/CHECKPOINT FREEZE	CEMT PERFORM SHUTDOWN IMM	No
/CHECKPOINT PURGE	CEMT PERFORM SHUTDOWN	No
/CHECKPOINT STATISTICS	CEMT PERFORM STATISTICS RECORD	Yes
/DBDUMP	None	Yes
/DBRECOVERY	None	Yes
/DELETE	None	Yes
/DEQUEUE	None	Yes
/DISPLAY ACTIVE	CEMT INQUIRE TASK	Yes
/DISPLAY CCTL	CEMT INQUIRE TASK	Yes
/DISPLAY DATABASE	None	Yes
/DISPLAY DBD	None	Yes
/DISPLAY POOL	None	Yes
/DISPLAY PSB	None	Yes
/ERESTART	SIT with START=AUTO resulting in EMERGENCY restart	No
/LOCK	None	Yes
/LOG	None	Yes
/MODIFY	None	No
/NRESTART CHECKPOINT 0	SIT START=INITIAL	No
/NRESTART (without CHECKPOINT 0)	SIT with START=AUTO resulting in WARM start	No
/PSTOP	None	Yes
/RMCHANGE	None	Yes
/RMDELETE	None	Yes
/RMGENJCL	None	Yes
/RMINIT	None	Yes
/RMLIST	None	Yes
/RMNOTIFY	None	Yes
/SSR	DSNC	No

Table 32 (Page 2 of 2). DBCTL Operator Commands and CICS Equivalents		
DBCTL Operator Command	**CICS Equivalent**	**Valid with CDBM**
/START DATABASE	None	Yes
/STOP DATABASE	None	Yes
/STOP THREAD	CEMT SET TASK PURGE	Yes
/SWITCH OLDS (IMS/ESA 4.1, or later)	Yes	Yes
/TRACE SET PI	None	Yes
/UNLOCK	None	Yes
/VUNLOAD (IMS/ESA Version 5.1 or later)	None	Yes
MVS MODIFY jobname,RECONNECT	CEMT PERFORM RECONNECT	N/A -- Use MVS command
MVS MODIFY jobname,RECONNECT	CEMT PERFORM RECONNECT	N/A --
MVS MODIFY jobname,STOP DUMP	CEMT PERFORM SHUTDOWN IMMEDIATE	N/A -- Use MVS command

Table 33. DBCTL Operator Commands and Keywords

DBCTL Operator Command	Keywords
/CHANGE	CCTL, PASSWORD, SUBSYS
/CHECKPOINT	FREEZE, PURGE, ABDUMP, SNAPQ
/DBDUMP	DATABASE
/DBRECOVERY	AREA, DATABASE
/DELETE	DATABASE, PASSWORD, PROGRAM
/DISPLAY	ACTIVE, AREA, CCTL, DATABASE, DBD, INDOUBT, MODIFY, OASN, SUBSYS, OLDS, POOL, PROGRAM, PSB, SHUTDOWN STATUS, STATUS, TRACE
/ERESTART	CHECKPOINT, COLDBASE, COLDCOMM, COLDSYS, FORMAT, NOBMP
/LOCK	DATABASE, PROGRAM
/LOG	None
/MODIFY	ABORT, COMMIT, PREPARE
/NRESTART	CHECKPOINT 0, FORMAT, NOPASSWORD, PASSWORD
/PSTOP	REGION
/RMCHANGE	DBRC modifier
/RMDELETE	DBRC modifier
/RMGENJCL	DBRC modifier
/RMINIT	DBRC modifier
/RMLIST	DBRC modifier
/RMNOTIFY	DBRC modifier
/SSR	Commands and keywords from appropriate subsystem (for example, DB2)
/START	AREA, AUTOARCH, DATABASE, OLDS, PROGRAM, REGION THREAD(1), WADS
/STOP	ADS, AREA, AUTOARCH, DATABASE, OLDS, PROGRAM, REGION THREAD(1), WADS
/SWITCH	OLDS
/TRACE	SET, MONITOR, PI, PSB, TABLE
/UNLOCK	DATABASE, PROGRAM
/VUNLOAD	AREA

Note: THREAD is a synonym for REGION

Appendix H. Available DBRC Commands for DBCTL

Table 34 lists the most commonly used DBRC commands for reference.

Table 34 (Page 1 of 2). DBRC Commands Available for DBCTL

Command	**Command**
BACKUP Command	
BACKUP.RECON	
CHANGE Commands	
CHANGE.ADS CHANGE.CA CHANGE.DB CHANGE.DBDSGRP CHANGE.PRILOG (for OLDS) CHANGE.PRILOG (for SLDS and TSLDS) CHANGE.RECON (for THT or REPTHT) CHANGE.SECLOG (for RLDS) CHANGE.SG CHANGE.UIC	CHANGE.BKOUT CHANGE.CAGRP CHANGE.DBDS CHANGE.IC CHANGE.PRILOG (for RLDS) CHANGE.RECON CHANGE.SECLOG (for OLDS) CHANGE.SECLOG (for SLDS and TSLDS) CHANGE.SUBSYS
DELETE Commands	
DELETE.ADS DELETE.BKOUT DELETE.CAGRP DELETE.DBDS DELETE.GSG DELETE.LOG (for OLDS) DELETE.RECOV DELETE.SG DELETE.UIC	DELETE.ALLOC DELETE.CA DELETE.DB DELETE.DBDSGRP DELETE.IC DELETE.LOG (for RLDS and SLDS) DELETE.REORG DELETE.SUBSYS
GENJCL Commands	
GENJCL.ARCHIVE GENJCL.CLOSE GENJCL.OIC GENJCL.RECOV	GENJCL.CA GENJCL.IC GENJCL.RECEIVE GENJCL.USER
INIT Commands	
INIT.ADS INIT.CAGRP INIT.DBDS INIT.GSG INIT.RECON	INIT.CA INIT.DB INIT.DBDSGRP INIT.IC INIT.SG
LIST Commands	
LIST.BKOUT LIST.DB LIST.DBDSGRP LIST.HISTORY LIST.LOG (for a category of records) LIST.SUBSYS	LIST.CAGRP LIST.DBDS LIST.GSG LIST.LOG (for a PRILOG family) LIST.RECON

Table 34 (Page 2 of 2). DBRC Commands Available for DBCTL	
Command	**Command**
NOTIFY Commands	
NOTIFY.ALLOC NOTIFY.CA NOTIFY.PRILOG (for OLDS) NOTIFY.PRILOG (for SLDS and TSLDS) NOTIFY.REORG NOTIFY.SECLOG (for RLDS) NOTIFY.SUBSYS	NOTIFY.BKOUT NOTIFY.IC NOTIFY.PRILOG (for RLDS) NOTIFY.RECOV NOTIFY.SECLOG (for OLDS) NOTIFY.SECLOG (for SLDS and TSLDS) NOTIFY.UIC
RESET Command	
RESET.GSG	

Appendix I. Year 2000 Considerations

I.1 Software Levels

The following software levels are Year 2000 compliant

- VSE/ESA 2.3
- CICS/VSE 2.3
- OS/390 2.4
- IMS/ESA 6.1
- CICS/TS 1.2
- COBOL/MVS
- LE/370

I.2 Rewrite Programs to Make Them Year 2000 Compliant

There are a number of areas users will need to look at for full Year 2000 compliance. This list is not exhaustive, but is designed to highlight common areas that need to be looked at in CICS/DL/I programs:

- Dates obtained using ACCEPT DATE statement in COBOL. Change all programs handling dates and times to obtain the date/time values from the LE/MVS library routines in Year-2000-compliant format. Although you do not need to complete your Year-2000 project before IMS/DBCTL is live, if you have to change programs anyway, it is easiest to make these other changes now. See section 8.8.11, "Rewrite Programs to Make Them Year 2000 Compliant" on page 130
- Date fields in programs. The year is often set as two digits. If you intend to keep it at two digits, then you need some logic to set a year number before which the year is considered to be in the new century. ***Document this well so that in 50 years time it can easily be changed.***
- Date fields in records stored on databases. This is more serious as the data could be there for a long time, and cause problems many years in the future. Correcting this may well need an UNLOAD/RELOAD, as the DBD will need changing.
- Programs reading log records. The format of these records will have changed in any case, so these need a complete rewrite. With OS/390 the timestamps may be in GMT. Check with your operations department as to how they did the initial program load the machine. You may also want to change some BMS maps to include four-digit year fields.
- Leap Year calculations. Remember that the year 2000 is a leap year.
- Century information in CICS. Changes have been made to enable you to code your applications to cope with the Year 2000. EXEC CICS FORMATTIME has additional options and all fields that have packed dates are altered.

I.3 Century Indicator in Packed Date Fields

All the fields containing the date in packed format are changed to show an additional indicator that represents the century. The format of the fields is changed to the following:

```
0CYYDDDs
```

where

C is the century indicator (0=1900s, 1=2000s).
YY is the year.
DDD is the day number
s is the sign indicator (positive).

The fields changed in this way on CICS control blocks data fields are:

- CSAJYDP
- CSASTYDP
- EIBDATE
- JCSRMDTE
- JCLRVCD
- JCLRDATE
- JCTVCD
- URDDATE

I.4 Extensions to EXEC CICS FORMATTIME Command

The EXEC CICS FORMATTIME command has new options added so that the four digits of the year may be returned to an application. The new keywords are:

- **DDMMYYYY(data-area)**.

  Specifies the 10-character user field where CICS is to return the date in day/month/year format (for example 17/06/1995).

- **MMDDYYYY(date-area)**.

  Specifies the 10-character user field where CICS is to return the date, in month/day/year format (for example 21/11/1995).

- **YYYYDDD(data-area)**.

  Specifies the 8-character user field where CICS is to return the date, in year/day format (for example 1995/200).

- **YYYYDDMM(data-area)**.

  Specifies the 10-character user field where CICS is to return the date, in year/day/month format (for example 1995/14/05).

- **YYYYMMDD(data-area)**.

  Specifies the 10-character user field where CICS is to return the date, in year/month/day format (for example 1995/10/21).

- **YYYYDDD(data-area)**

Specifies the 8-character user field where CICS is to return the date, in year/day format (for example 1995/200).

- **YYYYDDMM(data-area)**.

  Specifies the 10-character user field where CICS is to return the date, in year/day/month format (for example 1995/14/05).

- **YYYYMMDD(data-area)**.

  Sspecifies the 10-character user field where CICS is to return the date, in year/month/day format (for example 1995/10/21).

Appendix J. MVS Workload Manager

Every installation wants to make the best use of its resources, maintain the highest possible throughput, and achieve the best possible system responsiveness.

With MVS workload management, users define performance goals and assign a business importance rating to each goal. Users define the goals for work in business terms, and the system decides how much resource, such as CPU and storage, should be given to the work to meet its goal.

Manager of an installation should know what it is expected to accomplish in terms of performance goals, as well as how important it is to the business that each performance goal be achieved. With workload management, users define performance goals for work, and the system matches resources to the work to meet those goals, constantly monitoring and adapting processing to meet the goals. Reporting reflects how well the system is doing compared to its goals.

A service policy contains performance goals for all kinds of MVS-managed work expressed in the same terms. A service-level administrator identifies and categorizes all of an installation′s work and assigns the work performance goals in the workload management service policy. The services provide workload management with the information it needs to dynamically adapt to match resources to work to meet the performance goals.

Workload management matches system resources to meet the performance goal assigned to a service class. This management involves handling address space-related resources, such as processor storage, multiprogramming level (MPL), dispatching, and I/O queueing.

The services for subsystem work managers fall into the following categories:

- Work manager services
- Execution delay services
- Enclave services
- Queueing manager services
- Routing manager services
- Scheduling environment services
- Sysplex routing services
- Query system information service

Work manager services allow MVS to recognize:

- A subsystem work manager and the transactions it processes
- The service class goals associated with the transactions
- The address spaces that are processing the transactions

Based on this information, workload management (WLM) can determine whether goals are being met, and which work needs resources to meet the goals. The work manager services allow workload management to associate incoming work with a service class. When the work is associated with a service class, MVS knows the performance goal and importance level associated with the work, as well as understanding which address spaces are involved in processing the work request.

Defining WLM policy and setting other goals are beyond the scope of this book. Figure 150 shows a sample policy statement.

```
/******************************************************************/
/*                MEMBER = IEAICSDA                               */
/*                *****************                               */
/*                                                                */
/* COMMON IEAICSXX                     TIM EDWARDS   JANUARY 1990  */
/* UPDATE FOR HQ                       IAN DAWSON   29/01/90      HQ*/
/* UPDATE FOR AOF                      CHRIS GARROD 25/09/92     1CG*/
/* UPDATE CLASS=F INIT HOT BATCH       CHRIS GARROD 01/10/92     2CG*/
/*                                                                */
/* RPGN=14 FOR TSO                                                */
/* RPGN=15 FOR IMS CNTL/DLI (PLUS STCS IMSA, DLISAB + JOB IMSAARC)  */
/* RPGN=16 FOR IMS MPRS                                           */
/* RPGN=17 FOR IMS BMPS                                           */
/* RPGN=18 FOR N/W (VTAM, VAMP, VALIAS)                           */
/* RPGN=19 FOR CICS                                               */
/* RPGN=20 FOR DB2                                                */
/* RPGN=21 FOR BATCH                                              */
/* RPGN=22 FOR RMDS                                               */
/* RPGN=33 FOR APPC                                               */
/*  PGN=23 FOR COBRA CONTROLLED BATCH                             */
/* PUT CICS JOBS CLASS=C INTO PGN=11   TIM EDWARDS  25TH JAN 1991  */
/*  PGN=24 FOR HSM                     LUKE MARSHALL 17/12/92   @LM1*/
/*  PGN=9  FOR APPC                    LUKE MARSHALL 23/07/93   @LM2*/
/* RPGN=28 FOR SYSV                    LUKE MARSHALL 01/09/93   @LM3*/
/* RPGN=29 FOR NETV                    LUKE MARSHALL 01/09/93   @LM3*/
/* PGN=41  FOR LLA TO ISOLATE          LUKE MARSHALL 11/11/93   @LM4*/
/* PGN=42  FOR VLF TO ISOLATE          LUKE MARSHALL 11/11/93   @LM4*/
/* CICS TRACKING BY TXN CODE           CHRIS GARROD  28/04/94    1CI*/
/* RPGN=700-899 RESERVED FOR CICS                                1CI*/
/* PGN=1 FOR DB2ADGOV RPGN 601         LUKE MARSHALL 22/11/94   @LM5*/
/* PGN 112 113 FOR CICS OMEGAMON   LUKE MARSHALL 06/02/95       LM6*/
/* CICSCI, CICSOMEG AND CICSOC2 DELETED NICK VALLANCE 26/06/95     */
/******************************************************************/
/*                                                                */
/*     SEE INITIALISATION AND TUNING, SECTION HEADED : "CONTROL    */
/*     PERFORMANCE GROUPS FOR SYSTEM COMPONENT ADDRESS SPACES" FOR  */
/*     ADDRESS SPACES WHICH DON'T OBEY THE RULES                  */
/*                                                                */
/******************************************************************/
/*   STARTED TASK DEFINITIONS                                     */
/*                                                                */
/*   CHECK TASK NAMES FOR DB/DC TASKS                             */
/******************************************************************/
```

Figure 150 (Part 1 of 3). Sample IEAICSxx Member of SYS1.PARMLIB for Workload Management

```
 SUBSYS=STC,PGN=2
        TRXCLASS=HITASKS,   PGN=1
        TRXNAME=JES(1),     PGN=1,RPGN=699
        TRXNAME=LLA(1),     PGN=41      /*                      @LM4*/
        TRXNAME=VLF(1),     PGN=42      /*                      @LM4*/
        TRXNAME=VTAM(1),    PGN=1,RPGN=18
        TRXNAME=HSM(1),     PGN=24      /*                      @LM1*/
        TRXNAME=VAMP(1),    PGN=1,RPGN=18
        TRXNAME=DB2ADGOV(1),PGN=1,RPGN=601 /*                   @LM5*/
        TRXCLASS=MEDTASKS,  PGN=2
        TRXNAME=SMS(1),     PGN=2,RPGN=59
        TRXNAME=VALIAS(1),  PGN=2,RPGN=18
        TRXNAME=RMDS(1),    PGN=2,RPGN=22
        TRXNAME=SYSV(1),    PGN=2,RPGN=28 /*                    @LM3*/
        TRXNAME=NETV(1),    PGN=2,RPGN=29 /*                    @LM3*/
        TRXCLASS=LOWTASKS,  PGN=3
        TRXNAME=VPW(1),     PGN=3
        TRXNAME=IBT(1),     PGN=3
        TRXNAME=ADM(1),     PGN=3
        TRXNAME=COBRA(1),   PGN=3
        TRXNAME=COREMON,    PGN=3       /*                        HQ*/
        TRXNAME=HSMLOG(1),  PGN=3       /*                        HQ*/
        TRXNAME=IDDEWTR,    PGN=3       /*                        HQ*/
        TRXNAME=IVAMAPS,    PGN=3       /*                        HQ*/
        TRXNAME=IVBMAPS,    PGN=3       /*                        HQ*/
        TRXNAME=JOBWATCH,   PGN=3       /*                        HQ*/
        TRXNAME=NJEWTR,     PGN=3       /*                        HQ*/
        TRXNAME=PCMSWTR,    PGN=3       /*                        HQ*/
        TRXNAME=NVAS,       PGN=3
        TRXCLASS=IRLMTASK,  PGN=4
        TRXNAME=IRLMPROC,   PGN=4,RPGN=20
        TRXNAME=JRLMPROC,   PGN=4,RPGN=20
        TRXNAME=DB2CIRLM,   PGN=4,RPGN=20
        TRXNAME=DB2DIRLM,   PGN=4,RPGN=20
        TRXCLASS=DBDCTASK,  PGN=5
        TRXNAME=DB2(1),     PGN=5,RPGN=20
        TRXNAME=IMS(1),     PGN=5,RPGN=15 /* IMS CNTL REGION ONLY HQ*/
        TRXCLASS=DB2DIST,   PGN=6
        TRXNAME=DB2ADIST,   PGN=6,RPGN=20
        TRXNAME=DB2BDIST,   PGN=6,RPGN=20
        TRXNAME=DB2CDIST,   PGN=6,RPGN=20
        TRXNAME=DB2DDIST,   PGN=6,RPGN=20
        TRXNAME=CICS(1),    PGN=11,RPGN=19 /* CICS                HQ*/
        TRXNAME=CICS(5),    PGN=11,RPGN=19 /* CICS                HQ*/
        TRXNAME=DLI(1),     PGN=12,RPGN=15 /* IMS DLI DATABASE I/FHQ*/
        TRXNAME=BQFBMP(1),  PGN=13,RPGN=15 /* IMS BMP             HQ*/
        TRXNAME=DBR(1),     PGN=13,RPGN=15 /* IMS DBR  REG        HQ*/
        TRXNAME=MPR(1),     PGN=13,RPGN=16 /* IMS MPR'S           HQ*/
/*                                                                */
/******************************************************************/
/*   TSO DEFINITIONS                                              */
/******************************************************************/
 SUBSYS=TSO,PGN=7,RPGN=14
        USERID=XRMF(1),PGN=8          /* OPEX */
        USERID=STSGPF(1),PGN=8        /* PERF IDS*/
```

Figure 150 (Part 2 of 3). Sample IEAICSxx Member of SYS1.PARMLIB for Workload Management

```
/*******************************************************************/
/*   BATCH JOB AND CLASS DEFINITIONS                               */
/*                                                                 */
/*   CHECK JOBCLASSES                                              */
/*******************************************************************/
 SUBSYS=JES2,PGN=9,RPGN=21
        TRXNAME=IVTIM(1),PGN=5             /* TEST IMS CNTL        */
        TRXNAME=IMSTIM(1),PGN=5            /* TEST IMS CNTL        */
        TRXNAME=MPR(1),     PGN=13,RPGN=16 /* IMS MPR'S         1CG*/
        TRXNAME=AOF(1),PGN=13              /* TEST AOF          1CG*/
        TRXCLASS=F,PGN=13                  /* SPECIAL BATCH     2CG*/
        TRXCLASS=1,PGN=23                  /* COBRA CONTROLLED     */
        TRXCLASS=C,PGN=11,RPGN=19          /* CICS                 */
/*                                                                 */
/****************************************************************2LM*/
/*   APPC CLASS DEFINITIONS                                     2LM*/
/*                                                              2LM*/
/****************************************************************2LM*/
 SUBSYS=ASCH,PGN=9,RPGN=33             /*                       2LM*/
/*                                                                 */
/****************************************************************1CI*/
/*   CICS SUBSYSTEM  SYRA                                       1CI*/
/*                                                              1CI*/
/*                                                              1CI*/
/****************************************************************1CI*/
 SUBSYS=SYRA,RPGN=700                  /*                       1CI*/
        TRXNAME=CEMT,RPGN=701          /* CICS TRANSACTION      1CI*/
/*******************************************************************/
/*     END OF MEMBER = IEAICSDA                                   */
/*******************************************************************/
```

Figure 150 (Part 3 of 3). Sample IEAICSxx Member of SYS1.PARMLIB for Workload Management

Appendix K. IMS Monitor Reports

The IMS Monitor is the principal monitoring tool provided by the IMS product. It is an integral part of the control program in the DBCTL environment.

The IMS Monitor collects data while the DBCTL environment is operating. Information in the form of monitor records is gathered for all dispatch events and placed in a sequential data set. The IMS Monitor data set is specified on the IMSMON DD statement in the DBCTL region JCL; data is added to the data set when the /TRACE command activates the monitor. Here is an example:

```
/TRACE SET ON MONITOR ALL
```

The IMS product utility DFSUTR20 then format the data set and print the reports. The IMS/ESA product publication and ITSO redbook *IMS/ESA V5 Performance Guide, SG24-4637* have more detail information about the usage of the monitor and interpretation of reports. Figure 151 on page 302 is an example of the report. In general, the monitor report contains a large amount of data on IMS region performance; we selected only a few database-related reports.

```
DLI CALL REPORT                                                       00070000
DISTRIBUTION                                                          00080000
***I M S  M O N I T O R***  BUFFER POOL STATISTICS    TRACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0001

                  S Y S T E M  C O N F I G U R A T I O N


                                              SYSTEM CONFIGURATION       :  MVS/ESA
                                              IMS VERSION                :   5
                                              RELEASE LEVEL              :   1
                                              MODIFICATION NUMBER        :   0
***I M S  M O N I T O R***  BUFFER POOL STATISTICS    TRACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0002

                  M E S S A G E   Q U E U E   P O O L

                                                                        START TRACE        END   TRACE        DIFFERENCE
    NUMBER OF LOCATE CALLS FROM QMGR                                       104137             111768              7631
    NUMBER OF RECORD RELEASE CALLS FROM QMGR                                22979              25093              2114
    NUMBER OF LOCATE AND ALTER CALLS FROM QMGR                             130606             140512              9906
    NUMBER OF REQUESTS TO PURGE THE Q POOL                                     11                 12                 1
    NUMBER OF ADDRESS TO DRRN TRANSLATION REQUESTS                              0                  0                 0
    NUMBER OF READ REQUESTS                                                 14215              15672              1457
    NUMBER OF WRITE REQUESTS(TOTAL)                                          1510               1563                53
    NUMBER OF WRITES DONE BY PURGE                                            292                315                23
    NUMBER OF WAITS FOR PURGE COMPLETION                                        7                  8                 1
    NUMBER OF WAITS BECAUSE NO BUFFER AVAILABLE                                 0                  0                 0
    NUMBER OF WAITS FOR OTHER DECB TO READ THIS BUFFER                          1                  1                 0
    NUMBER OF WAITS FOR OTHER DECB TO WRITE THIS BUFFER                        11                 18                 7
    NUMBER OF WAITS FOR CONFLICTING END DEQ BUFFER REQ                          0                  0                 0
    NUMBER OF PSBS UNCHAINED FROM BUFFERS                                       0                  0                 0
    NUMBER OF CALLS TO QMGR.(TOTAL)                                        120627             134851             14224
    NUMBER OF CALLS TO REPOSITION A LOST BUFFER                                78                 79                 1
    NUMBER OF CALLS TO ENQ A MESSAGE                                        14573              15626              1053
    NUMBER OF CALLS TO DEQ ONE OR MORE MESSAGE                              11646              12431               785
    NUMBER OF CALLS TO CANCEL INPUT OR OUTPUT                                1252               1419               167

    QUOTIENT :  TOTAL NUMBER OF OSAM READS + OSAM WRITES + ALL IWAITS =     94.87
                _____________________________________________________
                         TOTAL NUMBER OF TRANSACTIONS
```

Figure 151 (Part 1 of 13). Sample IMS Monitor Reports

I M S M O N I T O R BUFFER POOL STATISTICS   TRACE START 1998 252 13:45:56   TRACE STOP   1998 252 14:17:17 PAGE 0008

V S A M   B U F F E R   P O O L

FIX INDEX/BLOCK/DATA	N/Y/N
SHARED RESOURCE POOL ID	XXXX
SHARED RESOURCE POOL TYPE	D

	START TRACE	END TRACE	DIFFERENCE
NUMBER OF RETRIEVE BY RBA CALLS RECEIVED BY BUF HNDLR	0	0	0
NUMBER OF RETRIEVE BY KEY CALLS	278	283	5
NUMBER OF LOGICAL RECORDS INSERTED INTO ESDS	0	0	0
NUMBER OF LOGICAL RECORDS INSERTED INTO KSDS	0	0	0
NUMBER OF LOGICAL RECORDS ALTERED IN THIS SUBPOOL	0	0	0
NUMBER OF TIMES BACKGROUND WRITE FUNCTION INVOKED	0	0	0
NUMBER OF SYNCHRONIZATION CALLS RECEIVED	1234	1389	155
NUMBER OF WRITE ERROR BUFFERS CURRENTLY IN THE SUBPOOL	0	0	0
LARGEST NUMBER OF WRITE ERRORS IN THE SUBPOOL	0	0	0
NUMBER OF VSAM GET CALLS ISSUED	278	283	5
NUMBER OF VSAM SCHBFR CALLS ISSUED	0	0	0
NUMBER OF TIMES CTRL INTERVAL REQUESTED ALREADY IN POOL	326	336	10
NUMBER OF CTRL INTERVALS READ FROM EXTERNAL STORAGE	31	31	0
NUMBER OF VSAM WRITES INITIATED BY IMS/ESA	0	0	0
NUMBER OF VSAM WRITES TO MAKE SPACE IN THE POOL	0	0	0
NUMBER OF VSAM READS FROM HIPERSPACE BUFFERS	0	0	0
NUMBER OF VSAM WRITES TO HIPERSPACE BUFFERS	0	0	0
NUMBER OF FAILED VSAM READS FROM HIPERSPACE BUFFERS	0	0	0
NUMBER OF FAILED VSAM WRITES TO HIPERSPACE BUFFERS	0	0	0

QUOTIENT : TOTAL NUMBER OF VSAM READS + VSAM WRITES =   0.00
____
TOTAL NUMBER OF TRANSACTIONS

Figure 151 (Part 2 of 13). Sample IMS Monitor Reports

IMS MONITOR ** GENERAL REPORTS ** RACE START 1998 252 13:45:56 TRACE STOP 1998 252 14:17:17 PAGE 0016

LATCH CONFLICT STATISTICS

LATCH NAMES	COUNT FIELD	AT START	AT END	DIFF.
DISP	CONTENTIONS	0	0	0
DCSL	CONTENTIONS	0	0	0
LUML	CONTENTIONS	0	0	0
CONV	CONTENTIONS	0	0	0
LUBT	CONTENTIONS	0	0	0
LRLT	CONTENTIONS	0	0	0
LBMP	CONTENTIONS	0	0	0
TCTB	CONTENTIONS	6	10	4
APSB	CONTENTIONS	0	0	0
PDRB	CONTENTIONS	0	0	0
PSBP	CONTENTIONS	0	0	0
DMBP	CONTENTIONS	0	0	0
PSBB	CONTENTIONS	0	0	0
DMBB	CONTENTIONS	0	0	0
PDRP	CONTENTIONS	0	0	0
DBAU	CONTENTIONS	0	0	0
DDRB	CONTENTIONS	0	0	0
DDRP	CONTENTIONS	0	0	0
DBBP	CONTENTIONS	0	0	0
DBLR	CONTENTIONS	0	0	0
SUBQ	CONTENTIONS	114	8	106
DBSL	CONTENTIONS	0	0	0
USER	CONTENTIONS	0	0	0
DBLT	CONTENTIONS	0	0	0
CCTL	CONTENTIONS	0	0	0
VLQB	CONTENTIONS	0	0	0
CBTS	CONTENTIONS	0	93	93
BLKM	CONTENTIONS	0	0	0
SMGT	CONTENTIONS	0	0	0
DBLK	CONTENTIONS	0	0	0
XCNQ	CONTENTIONS	0	0	0
ACTL	CONTENTIONS	0	5	5
LOGL	CONTENTIONS	8	3	5

Figure 151 (Part 3 of 13). Sample IMS Monitor Reports

```
    IMS MONITOR   ****REGION SUMMARY****            RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0019
                                         (A)                                    (B)
                                    ........ELAPSED TIME.........        NOT IWAIT TIME(ELAPSED-IWAIT)               DISTRIBUTION
               OCCURRENCES          TOTAL        MEAN     MAXIMUM        TOTAL        MEAN     MAXIMUM                   NUMBER
               ___________          _____        ____     _______        _____        ____     _______                  ______
SCHEDULING AND TERMINATION
__________ ___ ___________

SCHEDULE TO FIRST CALL
________ __ _____ ____
**REGION     3           1    35213779    35213779    35213779                                                              82
**REGION    11           1    37956070    37956070    37956070                                                              81
**REGION    12           1     4024702     4024702     4024702                                                              83
**REGION    39           1       25384       25384       25384                                                              84
**REGION    46           1   333377201   333377201   333377201                                                              73
**REGION    50           1   102749387   102749387   102749387                                                              80
**TOTALS                 6   513346523    85557753
ELAPSED EXECUTION
_______ _________
**REGION     3           1  1845330165  1845330165  1845330165                                                               4
**REGION    11           1  1842587874  1842587874  1842587874                                                               5
**REGION    12           1  1876519242  1876519242  1876519242                                                               6
**REGION    46           1  1547166743  1547166743  1547166743                                                               2
**REGION    50           1  1777794557  1777794557  1777794557                                                               3
**TOTALS                 5  8889398581  1777879716
DL/I CALLS                                                                                         IWT/CALL(C)
____ _____                                                                                         ___________
**REGION     3         279      568384        2037      118515      568384        2037      118515        0.00           13A,B,C
**REGION     6         164      870755        5309      159956      411557        2509       46440        0.21           39A,B,C
**REGION     7          29      101465        3498       84099      101465        3498       84099        0.00           55A,B,C
**REGION    11          22       28531        1296        5860       28531        1296        5860        0.00           12A,B,C
**REGION    12         414     3401975        8217      893693     3279804        7922      893693        0.01            9A,B,C
**REGION    14         793     1392519        1756       75781     1318667        1662       44244        0.00           58A,B,C
**REGION    17         691     1704444        2466       99681     1585966        2295       99681        0.03           19A,B,C
**REGION    22          79     1753921       22201      200128     1753921       22201      200128        0.00           23A,B,C
**REGION    24         364      939042        2579       44077      931665        2559       44077        0.00           11A,B,C
**REGION    26        2185     4010862        1835      655670     2762293        1264      624453        0.02           25A,B,C
**REGION    29         243     7143405       29396      418091     7130259       29342      418091        0.01           10A,B,C
```

Figure 151 (Part 4 of 13). Sample IMS Monitor Reports

```
**REGION    31          40       19286         482        5841       10141         253        1014        0.05            70A,B,C
**REGION    34         238      716101        3008       61986      325758        1368       29294        0.13            37A,B,C
**REGION    35          48      128342        2673       65317      118866        2476       65317        0.04            53A,B,C
**REGION    38         993     2527052        2544      273329     2260343        2276      273329        0.02            15A,B,C
**REGION    39         804     2790689        3471      294706     2083470        2591      175479        0.03            18A,B,C
**REGION    40          18        1715          95         205        1715          95         205        0.00            69A,B,C
**REGION    42          35      293490        8385       74425      261560        7473       74425        0.05            27A,B,C
**REGION    46           4       12478        3119        5339       12478        3119        5339        0.00            72A,B,C
**REGION    47        3705    20405376        5507     1809249     8701389        2348     1341915        0.19             7A,B,C
**REGION    50         162      830172        5124      102502      830172        5124      102502        0.00            56A,B,C
**REGION    52          30     1232871       41095      987516     1026129       34204      835528        0.36            24A,B,C
**REGION    54          63      608252        9654      124052      608252        9654      124052        0.00            30A,B,C
**TOTALS             11403    51481127        4514                36112785        3166                    0.08
DD1A     SERVICE AND COMMAND CALLS

     IMS MONITOR   ****REGION SUMMARY****              RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0020
**REGION     7           5     1091738      218347      984260                                                            450A
**REGION    17          78     1526178       19566       92695                                                            442A
**REGION    22          61     1330284       21807      174053                                                            443A
**REGION    24          64     1094564       17102      110661                                                            437A
**REGION    26          88     3874034       44023      627952                                                            446A
**REGION    29         154     4617513       29983      269837                                                            436A
**REGION    31           6    33793827     5632304    33706029                                                            453A
**REGION    35           6      326176       54362      246088                                                            449A
**REGION    38         196     3131940       15979      241807                                                            441A
**REGION    39         215     3427778       15943      187225                                                            438A
**REGION    40           8      267819       33477      125376                                                            452A
**REGION    42          34      967301       28450      192790                                                            445A
**REGION    47         145     1673765       11543      146143                                                            439A
**REGION    52          18      407374       22631       79299                                                            444A
**REGION    54          35     1065112       30431      149948                                                            447A
**TOTALS              1113    58595403       52646
```

Figure 151 (Part 5 of 13). Sample IMS Monitor Reports

```
DD1A     NORMAL CALLS
________ _______ ____
**REGION     7        103    22226559      215791     2928839                                                           450B
**REGION    17        119     3743724       31459     2238350                                                           442B
**REGION    22        285     2412590        8465     1020591                                                           443B
**REGION    24        110     2141945       19472      577152                                                           437B
**REGION    26       1760   103792783       58973    17867718                                                           446B
**REGION    29        322     1381041        4288       66260                                                           436B
**REGION    35         10     5045629      504562     2080932                                                           449B
**REGION    38       1739    15510426        8919     1414235                                                           441B
**REGION    39       1652    34714878       21013     3434493                                                           438B
**REGION    40         26     3893841      149763     1044565                                                           452B
**REGION    42         49    14069547      287133     2353476                                                           445B
**REGION    47       1051    13551495       12893     2451518                                                           439B
**REGION    52        243    18341544       75479     3318308                                                           444B
**REGION    54       1799    25073461       13937     3438566                                                           447B
**TOTALS             9268   265899463       28690
MD1Q     SERVICE AND COMMAND CALLS
________ _______ ___ _______ _____
**REGION    38          8       46530        5816       21879                                                           448A
**REGION    39         16      145208        9075       58251                                                           451A
**REGION    47         80      576299        7203       93869                                                           440A
**TOTALS              104      768037        7384
MD1Q     NORMAL CALLS
________ _______ ____
**REGION    38         16       16904        1056        2909                                                           448B
**REGION    39         32       46088        1440        5988                                                           451B
**REGION    47        160      266301        1664       27111                                                           440B
**TOTALS              208      329293        1583
IDLE FOR INTENT
____ ___ ______
                     NONE
CHECKPOINT
__________           NONE
```

Figure 151 (Part 6 of 13). Sample IMS Monitor Reports

```
      IMS MONITOR   ****REGION IWAIT****               RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0061
                                    .........IWAIT TIME..........                       DISTRIBUTION
**REGION    39 OCCURRENCES       TOTAL        MEAN     MAXIMUM       FUNCTION  MODULE      NUMBER
  ______    ___________         _____        ____     _______       ________  ______      ______
SCHEDULING + TERMINATION
__________  ___________
DL/I CALLS
____ _____
                          24      466971       19457      265822    DD=RWBRWKD    VBH            21
                           6      178418       29736       41738    DD=RWFHDP     VBH            22
                           1        3349        3349        3349    DD=SHMSG      QMG            51
                           1        2773        2773        2773    DD=SHMSG1     QMG            52
                           4      107286       26821       34767    DD=RWFHDX     VBH            64
___TOTAL___
   _____                  36      758797       21077
DD1A     CALLS
________ _____
                          51      244564        4795       14400   TERM THRD      D50
                          49      433676        8850      107010   COMMIT PH.1    P10
                        1652    34714878       21013     3434493   NORMAL CALL    PR0
                          51     1095028       21471      187225   CREATE THRD    CT0
                          51     1462188       28670       63104   SIGNON         S00
                           3       61155       20385       56169   ABORT          AB0
                           6      131096       21849       38867   COMMIT PH.2    P20
                           4          71          17          18   TERM THRD      P30
___TOTAL___
   _____                1867    38142656       20429
MD1Q     CALLS
________ _____
                           4      132946       33236       58251   TERM THRD      D50
                           4        4412        1103        1118   COMMIT PH.1    P10
                          32       46088        1440        5988   NORMAL CALL    PR0
                           4         326          81          91   CREATE THRD    CT0
                           4        7524        1881        3243   SIGNON         S00
___TOTAL___
   _____                  48      191296        3985
```

Figure 151 (Part 7 of 13). Sample IMS Monitor Reports

```
    IMS MONITOR   ****PROGRAMS BY REGION****        RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0077
                                             (A)                                   (B)
                                  ELAPSED EXECUTION TIME            SCHEDULING END TO FIRST CALL      DISTRIBUTION
             OCCURRENCES        TOTAL        MEAN     MAXIMUM       TOTAL        MEAN     MAXIMUM        NUMBER

             ___________        _____        ____     _______       _____        ____     _______        ______
**REGION     3
________   _ 3
    RRDQEAR            1  1845330165  1845330165  1845330165    35213779    35213779    35213779          354A,B
 REGION TOTALS         1  1845330165  1845330165                35213779    35213779
**REGION    11
________   _11
    RTBMATA            1  1842587874  1842587874  1842587874    37956070    37956070    37956070          358A,B
 REGION TOTALS         1  1842587874  1842587874                37956070    37956070
**REGION    46
________   _46
    AOFWFI             1  1547166743  1547166743  1547166743   333377201   333377201   333377201          346A,B
 REGION TOTALS         1  1547166743  1547166743               333377201   333377201
**REGION    50
________   _50
    EVISPPII           1  1777794557  1777794557  1777794557   102749387   102749387   102749387          350A,B
 REGION TOTALS         1  1777794557  1777794557               102749387   102749387

    IMS MONITOR   ****PROGRAM SUMMARY****           RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0080
                                                           (A)........(B)........          (A)........(B)........
                                                     I/O   TRAN.       CPU        .        ELAPSED    SCHED.TO    .        ELAPSED
         NO.    TRANS.           CALLS      I/O    IWAITS  DEQD.      TIME     DISTR.       TIME      1ST CALL  DISTR.       TIME
PSBNAME  SCHEDS.  DEQ.     CALLS  /TRAN    IWAITS  /CALL  /SCH.     /SCHED.     NO.       /SCHED.     /SCHED.    NO.       /TRANS.
_______  _______  ____     _____  _____    ______  _____  _____     _______    ___       _______     _______    ___       _______

AOFWFI        1      2         4    2.0         0    0.0    2.0       10010  345A,B  1547166743   333377201  347A,B   773583371
EVISPPII      1     13       162   12.4         0    0.0   13.0       10010  349A,B  1777794557   102749387  351A,B   136753427
RRDQEAR       1      0       279    -           0    0.0    0.0       10010  353A,B  1845330165    35213779  355A,B           -
RTBMATA       1      0        22    -           0    0.0    0.0       10010  357A,B  1842587874    37956070  359A,B           -
RRPA000       1      1       414  414.0         7    0.0    1.0       10010  361A,B  1876519242     4024702  363A,B  1876519242
AGEYYOP       0      0      2719    -        1946    0.7    0.0           0  428A,B          18           0  343A,B           -
**TOTALS      5     16      3600  225.0      1953    0.5    3.2       10010          2153983428   102669304            673119821
```

Figure 151 (Part 8 of 13). Sample IMS Monitor Reports

IMS MONITOR ****PROGRAM I/O**** RACE START 1998 252 13:45:56 TRACE STOP 1998 252 14:17:17 PAGE 0081

PSBNAME	PCB NAME	IWAITS	.........IWAIT TIME.......... TOTAL	MEAN	MAXIMUM	DDN/FUNC	MODULE	DISTR. NO.
PREG0047	I/O PCB	343	2153866	6279	120074	RWBRWKD	VBH	364
		3	10292	3430	3640	SHMSG1	QMG	388
		5	24172	4834	8948	SHMSG	QMG	396
		1	2557	2557	2557	QBLKS	QMG	408
		3	38382	12794	26989	MB216A	VBH	411
		3	87446	29148	55243	RWFHDP	VBH	415
		1	5104	5104	5104	RWFHDX	VBH	416
		2	8012	4006	4500	MB218	VBH	420
		1	3266	3266	3266	MB218X	VBH	421
		1	3148	3148	3148	MB218S1	VBH	422
	PCB TOTAL							
		363	2336245	6435				
	RWBRWKD	83	1484755	17888	63357	RWBRWKD	VBH	365
	PCB TOTAL							
		83	1484755	17888				
	RWFHDP	182	5619066	30873	83063	RWFHDP	VBH	366
		13	171561	13197	40721	RWFHDX	VBH	399
	PCB TOTAL							
		195	5790627	29695				
	RWBRDED	61	1552792	25455	53429	RWBRDED	VBH	370
	PCB TOTAL							
		61	1552792	25455				
	RWBRTAD	2	41542	20771	21661	RWBRTAD	VBH	398
	PCB TOTAL							
		2	41542	20771				
	MB218	15	181263	12084	26164	MB218S1	VBH	423
		4	83742	20935	27848	MB218	VBH	424
		15	202329	13488	46525	MB218X	VBH	425

Figure 151 (Part 9 of 13). Sample IMS Monitor Reports

```
         PCB TOTAL
         ___ _____        34      467334       13745
          MB322            1       30692       30692       30692   MB222A     VBH          426

         PCB TOTAL
         ___ _____         1       30692       30692
          DD1A            31      146327        4720       20652  TERM THRD   D50
                          31       64450        2079       17089  COMMIT PH.1 P10
                          20         453          22         143  TERM THRD   P30
                        1051    13551495       12893     2451518  NORMAL CALL PR0
                          31      329360       10624      146143  CREATE THRD CT0
                          31     1117644       36053       71964  SIGNON      S00
                           1       15531       15531       15531  COMMIT PH.2 P20
      IMS MONITOR   ****PROGRAM I/O****                RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0082
                                      .........IWAIT TIME..........                          DISTR.
PSBNAME  PCB NAME      IWAITS        TOTAL        MEAN     MAXIMUM   DDN/FUNC  MODULE        NO.

_______  ___ ____      ______       _____        ____     _______   ________  ______        ___


      SUBSYS TOTAL
      ______ _____      1196    15225260       12730
          MD1Q            20      510464       25523       93869  TERM THRD   D50
                          20       24200        1210        4997  COMMIT PH.1 P10
                         160      266301        1664       27111  NORMAL CALL PR0
                          20       11082         554        5803  CREATE THRD CT0
                          20       30553        1527        3452  SIGNON      S00

      SUBSYS TOTAL
      ______ _____       240      842600        3510
PSB TOTAL
___ _____               2175    27771847       12768
GRAND TOTAL
_____ _____            11722   341404190       29125
```

Figure 151 (Part 10 of 13). Sample IMS Monitor Reports

```
     IMS MONITOR   ****TRANSACTION QUEUING****         RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0092
                                                       (B)                    (A)
                   NUMBER      NUMBER      ..ON QUEUE WHEN SCHEDULED.....   DEQUED      DISTRIBUTION
TRANSACTION        DEQUED      SCHEDS.     MINIMUM    MEAN        MAXIMUM    MEAN        NUMBER
___________        ______      ______      _______    ____        _______    ____        ______

 AOFL                 2           1           0       0.00           0       2.00         344A,B
 EVITPPI1            13           1           0       0.00           0      13.00         348A,B
 8501                 1           1           0       0.00           0       1.00         360A,B

     IMS MONITOR   ****REPORTS****                     RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0093
NO INTENT FAILURES IN THIS TRACE
NO POOL SPACE FAILURES IN THIS TRACE
NO DEADLOCK EVENTS IN THIS TRACE
TOTAL TIMES ECBS WAITED FOR SAPS =       0
MONITOR OVERHEAD DATA                                                                                                              I
_____________________                                                                                                              I
    1880543 MILLISECONDS, TRACE INTERVAL
      73375 MONITOR RECORDS WERE PRODUCED
```

Figure 151 (Part 11 of 13). Sample IMS Monitor Reports

```
     IMS MONITOR   **RUN PROFILE**                    RACE START  1998 252  13:45:56    TRACE STOP   1998 252  14:17:17  PAGE 0094
TRACE ELAPSED TIME IN SECONDS......1880.5
TOTAL NUMBER OF MESSAGES DEQUEUED......16
TOTAL NUMBER OF SCHEDULES        .......5
NUMBER OF TRANSACTIONS PER SECOND    0.0
TOTAL NUMBER OF DL/I CALLS ISSUED.....1685
NUMBER OF DL/I CALLS PER TRANSACTION  105.3
NUMBER OF SERVICE CALLS TO SUBSYSTEM DD1A    ....1113
NUMBER OF SERVICE CALLS PER TRANSACTION   69.5
NUMBER OF COMMAND CALLS TO SUBSYSTEM DD1A    .......0
NUMBER OF NORMAL  CALLS TO SUBSYSTEM DD1A    ....9268
NUMBER OF NORMAL  CALLS PER TRANSACTION  579.2
NUMBER OF SERVICE CALLS TO SUBSYSTEM MD1Q    .....104
NUMBER OF SERVICE CALLS PER TRANSACTION    6.5
NUMBER OF COMMAND CALLS TO SUBSYSTEM MD1Q    .......0
NUMBER OF NORMAL  CALLS TO SUBSYSTEM MD1Q    .....208
NUMBER OF NORMAL  CALLS PER TRANSACTION   13.0
NUMBER OF OSAM BUFFER POOL I/O'S.           1,    0.0 PER TRANSACTION
NUMBER OF MESSAGE QUEUE POOL I/O'S.......1518,   94.8 PER TRANSACTION
NUMBER OF FORMAT BUFFER POOL I/O'S.........24,    1.5 PER TRANSACTION
RATIO OF PROGRAM ELAPSED TO DL/I ELAPSED:
              REGION   3: 246.53
              REGION  11: 652.20
              REGION  12: 551.59
              REGION  50: 141.40
RATIO OF DL/I ELAPSED TO DL/I IWAIT:
              REGION   6:   1.89
              REGION  12:  27.83
              REGION  14:  18.84
              REGION  17:  14.38
              REGION  24: 126.89
              REGION  26:   3.21
              REGION  29: 545.29
              REGION  31:   2.11
              REGION  34:   1.83
              REGION  35:  13.50
              REGION  38:   9.47
              REGION  52:   5.96
```

Figure 151 (Part 12 of 13). Sample IMS Monitor Reports

IMS MONITOR ****CALL SUMMARY**** RACE START 1998 252 13:45:56 TRACE STOP 1998 252 14:17:17 PAGE 0096

PSB NAME	PCB NAME	CALL FUNC	LEV NO.SEGMENT	STAT CODE	CALLS	IWAITS	(C) IWAITS/ CALL	(A) ..ELAPSED TIME... MEAN	MAXIMUM	(B) .NOT IWAIT TIME.. MEAN	MAXIMUM	DISTRIB. NUMBER
PREG0047	I/O PCB	GU	( )	QC	189	357	1.88	25055	175808	12867	149265	85A,B,C
		ISRT	( )		402	3	0.00	1792	53598	1753	53598	87A,B,C
		GN	( )		185	0	0.00	151	1879	151	1879	89A,B,C
		CHNG	( )		20	0	0.00	2590	14803	2590	14803	150A,B,C
		GU	( )		1	3	3.00	27622	27622	10741	10741	165A,B,C
		GN	( )	QD	13	0	0.00	30	38	30	38	217A,B,C
		PURG	( )		6	0	0.00	414	493	414	493	321A,B,C
		I/O PCB SUBTOTAL			816	363	0.44	6821		3958		
	RWBRWKD	ISRT	(01)RTSENR1		113	83	0.73	16164	65618	3024	15408	86A,B,C
		DLET	(01)RTSENR1		115	0	0.00	2458	20659	2458	20659	90A,B,C
		GN	(01)RTSENR1	GE	115	0	0.00	563	9714	563	9714	91A,B,C
		GU	(01)RTSENR1		115	0	0.00	774	7445	774	7445	92A,B,C
		DL/I PCB SUBTOTAL			458	83	0.18	4941		1699		
	RWFHDP	GN	(01)ROOTSEGM		871	178	0.20	7665	88979	1544	51497	94A,B,C
		GU	(01)ROOTSEGM		118	13	0.11	4723	91316	1579	12678	95A,B,C
		REPL	(01)ROOTSEGM		1	0	0.00	559	559	559	559	307A,B,C
		ISRT	(01)ROOTSEGM		1	4	4.00	111185	111185	22665	22665	308A,B,C
		GU	(00)	GE	1	0	0.00	1403	1403	1403	1403	309A,B,C
		DL/I PCB SUBTOTAL			992	195	0.19	7406		1568		
	RWBRDED	GU	(00)	GE	8	2	0.25	12367	54859	647	1430	96A,B,C
		GNP	(02)RTSSAA2		108	14	0.12	2711	41711	550	7303	97A,B,C
		GNP	(02)RTSSAA3		110	1	0.00	402	4204	373	2813	98A,B,C
		GU	(01)RTSSAA1		216	41	0.18	5889	80270	584	6001	99A,B,C
		GNP	(01)RTSSAA1	GE	106	2	0.01	731	28667	438	9840	135A,B,C
		GNP	(02)RTSSAA4		2	1	0.50	23608	46632	954	1324	164A,B,C
		DL/I PCB SUBTOTAL			550	61	0.11	3332		509		
	RWFHDX	GU	(01)SEKINDEX		8	0	0.00	755	1590	755	1590	100A,B,C
		DL/I PCB SUBTOTAL			8	0	0.00	755		755		
	MB322	GU	(01)M570		363	1	0.00	832	32135	748	14697	151A,B,C

Figure 151 (Part 13 of 13). Sample IMS Monitor Reports

Appendix L. IMSPARS Reports

IMSPARS is a performance analysis and tuning aid for IMS/ESA. IMSPARS analyzes data from the IMS log, providing comprehensive information about:

- Transaction transit times (actual system performance time)
- IMS resource use
- IMS resource availability

The following are a few sample reports for reference. Refer to product publications for detailed information.

L.1 Sample JCL to Run IMSPARS

Figure 152 shows a JCL sample to print IMSPARS reports.

```
//IMSPARS  JOB (000),'IMSAB4',CLASS=L,MSGCLASS=H,MSGLEVEL=(1,1),
//         REGION=0M
//   SET SYS1=UKIMSAB4
//*********************************************************************
//*
//*  EXECUTE IMSPARS
//*
//*********************************************************************
//IMSPARS  EXEC PGM=GPAR,REGION=0M
//STEPLIB    DD DISP=SHR,DSN=IMS.UKIMSAB4.PGMLIB
//SYSPRINT   DD SYSOUT=*,DCB=BLKSIZE=6118
//CMDLIB     DD DUMMY
//MGRXDD     DD SYSOUT=*,DCB=BLKSIZE=6118
//RPTOUT     DD SYSOUT=*,DCB=BLKSIZE=6118
//LOGDD      DD SYSOUT=*,DCB=BLKSIZE=6118
//DCTRDD     DD SYSOUT=*,DCB=BLKSIZE=6118
//SPRTAVGQ   DD DUMMY,DCB=(RECFM=VB,LRECL=4092,BLKSIZE=32760)
//SPRTAVGO   DD DUMMY,DCB=(RECFM=VB,LRECL=4092,BLKSIZE=32760)
//SPRTEXPQ   DD DUMMY,DCB=(RECFM=VB,LRECL=4092,BLKSIZE=32760)
//SPRTBLDQ   DD DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=23440)
//LOGIN      DD DSN=IMS.UKIMSAB4.SLDSS.D98127.T1630190.V89,
//           DCB=BUFNO=40,DISP=SHR
//           DD DSN=IMS.UKIMSAB4.SLDSS.D98127.T1659049.V41,
//           DCB=BUFNO=40,DISP=SHR
//SYSIN      DD *
*
* CONTROL STATEMENTS FOR IMSPARS
*
 PARM NOSTAE
 IMSPARS MGREX
 IMSPARS AVAIL(DATABASE,PROGRAM)
 IMSPARS CPUR
 IMSPARS DBUPDATE
 IMSPARS EXEC
//
```

Figure 152. Sample JCL to Print IMSPARS Reports

L.2 Sample Reports from IMSPARS

Figure 153 on page 317 shows a sample IMSPARS exception report. Figure 154 on page 318 shows a sample IMSPARS of log tape system messages.

```
HH.MM.SS.TH ERROR     PROGRAM  TRANCO       1ST LOG REC      LAST LOG REC
___________ ________  ________ ______ ________________ ________________
16.05.19.00 ABND0777  JBDS04P  JBDS04 000000000353B811 000000000353B83A
16.05.24.00 ABND0777  JBD004P  JBD004 000000000353BA23 000000000353BA58
16.06.51.50        8           VA4105 000000000353E167
16.13.22.90        8           CF1478 000000000355780C
16.13.40.70      128           CF1478 0000000003558C69
16.18.15.10        8           CF3209 0000000003566B2C
16.30.39.70 ABND0777  JBD008P  JBD008 00000000035D3DE3 00000000035D3E17
16.30.44.70 ABND0777  JBD008P  JBD008 00000000035D4E56 00000000035D4E87
16.30.49.70 ABND0777  JBD008P  JBD008 00000000035D62BC 00000000035D62ED
16.30.54.70 ABND0777  JBD008P  JBD008 00000000035D73CB 00000000035D73FC
16.35.56.80        8           CF3048 00000000035ED0A0
16.35.59.70 ABND0777  JBD008P  JBD008 00000000035ED195 00000000035ED1C6
16.39.10.00        8           IT2399 00000000035F3033
16.40.45.00        8           IT3148 00000000035F6D57
16.55.20.40 ABND0777  JBD009P  JBD009 0000000003639AB4 0000000003639B46
16.55.20.40 PSAB0777  JBD009P  JBD009 0000000003639B97
16.55.30.30 ABND0777  JBD009P  JBD009 000000000363CFB8 000000000363CFE5


START  5/07/1998  15.58.07.10              I M S    L O G   T A P E   R
HH.MM.SS.TH ERROR     PROGRAM  TRANCO       1ST LOG REC      LAST LOG REC
___________ _________ ________ ______ ________________ ________________
17.17.36.00 ABND0777  JBDS03P  JBDS03 00000000036B8AD6 00000000036B8AFF
17.17.41.00 ABND0777  JBD003P  JBD003 00000000036B8C7B 00000000036B8CA8
17.17.51.00 ABND0777  JBD003P  JBD003 00000000036B90C5 00000000036B90F2
17.19.36.10 ABND0777  JBD003P  JBD003 00000000036BA9BB 00000000036BA9EC
17.24.48.80      128           CSIA   00000000036C3BC5
17.27.46.20 ABND0777  JBDS04P  JBDS04 00000000036C8016 00000000036C803F
17.27.51.30 ABND0777  JBD004P  JBD004 00000000036C829B 00000000036C82CC
17.28.00.80 ABND0777  JBDS04P  JBDS04 00000000036C87DA 00000000036C8803
17.28.06.30 ABND0777  JBD004P  JBD004 00000000036C89A7 00000000036C89D4
17.28.16.10 ABND0777  JBD004P  JBD004 00000000036C8C3E 00000000036C8C6A
17.28.21.30 ABND0777  JBDS04P  JBDS04 00000000036C8F49 00000000036C8F72
17.28.26.20 ABND0777  JBD004P  JBD004 00000000036C90B8 00000000036C90E4
17.28.35.20 ABND0777  JBD004P  JBD004 00000000036C95F2 00000000036C961F
17.28.41.10 ABND0777  JBDS04P  JBDS04 00000000036C9830 00000000036C9859
17.28.45.80 ABND0777  JBD004P  JBD004 00000000036C992E 00000000036C9965
17.28.56.10 ABND0777  JBD004P  JBD004 00000000036C9B07 00000000036C9B34
17.29.01.00 ABND0777  JBDS04P  JBDS04 00000000036C9BC0 00000000036C9BE9


START  5/07/1998  15P E   R E P O R T (V1M9)
                    DITIONS
                    ______________

HH.MM.SS.TH ERROR     PROGRAM  TRANCO       1ST LOG REC      LAST LOG REC
___________ _________ ________ ______ ________________ ________________
17.29.06.30 ABND0777  JBD004P  JBD004 00000000036C9D92 00000000036C9DBF
17.29.16.40 ABND0777  JBD004P  JBD004 00000000036CA20B 00000000036CA238
17.29.21.40 ABND0777  JBDS04P  JBDS04 00000000036CA67E 00000000036CA6A7
17.36.29.00 ABND0777  JBD005P  JBD005 00000000036E7CC3 00000000036E7CF8
17.36.41.00 ABND0777  JBD005P  JBD005 00000000036E7FF8 00000000036E802D
17.39.35.20        8           JA9315 00000000036EA4EA
17.40.31.60 ABND0777  JBD005P  JBD005 00000000036EC981 00000000036EC9B6
17.43.24.90        8           US1781 00000000036F0F24
```

Figure 153. Sample IMSPARS Exception Report

I M S L O G T A P E R E P O R T S (V1M9)
S Y S T E M M E S S A G E S

```
   5/07/98 16.30.18.90        LOGIN      IMS STARTED
   5/07/98 16.30.18.90        LOGIN    REPORTING STARTS, WITHIN SPECIFIED TIME RANGE
START  5/07/1998  16.30.18.90            I M S   L O G   T A P E   R E P O R T S (V1M9) END  5/07/1998 17.46.38.40 PAGE      1
```

CPU USAGE REPORT

	REGN	PROGRAM	TRANCODE	OBSERV COUNT	*******CPU TIME (MSECS)******* TOTAL	MEAN	SDEV	ELAPSE /CPU RATIO	******ELAPSE TIME (MSECS)***** TOTAL	MEAN	SDEV	TRANS /OBS
>	1	NOTAVAIL	NOTAVAIL	234	0	0			4,579,500	19,571	.00	1.00
SUBTOTAL				234	0	0			4,579,500	19,571	.00	1.00
	2	LOSATPR	9180	49	10	0	.00	139.64	1,400	29	.00	1.00
	2	RRPP650	*BLANK*	2	20	10	.00	9.97	200	100	.00	.00
	2	RTBMANA	*BLANK*	1	10			784.94	58,000			.00
>	2	RTBMATA	9171	280	0	0			2,997,100	10,704	.00	1.00
	2	RTBMATE	*BLANK*	1	10			738.08	7,400			.00
SUBTOTAL				333	50	0	7.25	122.84	3,064,100	9,202	.51	.99
	3	JBDA08P	JBDA08	2	29	15	.01	3.40	100	50	1.00	1.00
	3	JBDC08P	JBDC08	1	105			2.87	300			1.00
>	3	JBDJ00P	JBDJ00	99	43,587	440	.75	8.80	383,600	3,875	.76	1.00
	3	JBDL08P	JBDL08	30	3,571	119	.61	39.65	141,600	4,720	1.38	1.00
	3	JBDO08P	JBDO08	68	15,526	228	.26	4.57	70,900	1,043	.88	1.00
	3	JBDS08P	JBDS08	65	1,181	18	.14	28.55	33,700	518	.93	1.00
	3	JBDW00P	JBDW00	1	9			.00	0			1.00
SUBTOTAL				266	64,007	241	1.11	9.85	630,200	2,369	1.41	1.00
	6	RWYMQRP	RWYMQRT	38	1,669	44	.05	2.34	3,900	103	.42	1.00
	6	RWYTC3P	RWYTC3T	38	1,639	43	.07	2.81	4,600	121	.99	1.00
	6	RWYTC4P	RWYTC4T	38	1,005	26	.07	3.58	3,600	95	.64	1.00
SUBTOTAL				114	4,313	38	.22	2.81	12,100	106	.77	1.00

Figure 154 (Part 1 of 5). Sample IMSPARS Reports

START 5/07/1998 16.30.18.90 I M S L O G T A P E R E P O R T S (V1M9) END 5/07/1998 17.46.38.40 PAGE 22

CPU USAGE REPORT

REGN	PROGRAM	TRANCODE	OBSERV COUNT	*******CPU TIME (MSECS)******* TOTAL	MEAN	SDEV	ELAPSE /CPU RATIO	******ELAPSE TIME (MSECS)***** TOTAL	MEAN	SDEV	TRANS /OBS
152	JBDC04P	JBDC04	1	139			5.03	700			1.00
152	JBDF04P	JBDF04	2	128	64	.00	3.92	500	250	.00	1.00
152	JBD001P	JBD001	1	190			3.16	600			1.00
> 152	JBD002P	JBD002	1	0				227,500			1.00
152	JBD003P	JBD003	6	1,660	277	.53	10.24	17,000	2,833	.95	1.00
152	JBD004P	JBD004	9	1,495	166	.60	2.81	4,200	467	.40	1.00
152	JBD005P	JBD005	5	428	86	.30	3.27	1,400	280	.04	1.00
152	JBDQ01P	JBDQ01	1	9			.00	0			1.00
152	JBDS02P	JBDS02	8	146	18	.00	17.14	2,500	313	.00	1.00
SUBTOTAL			100	14,670	147	1.01	30.84	452,400	4,524	4.99	1.00
155	RWBRSMP	RWBSDTS	33	446	14	.16	6.06	2,700	82	1.02	1.00
155	SP501	COMSEC	5	50	10	.05	6.03	300	60	1.33	1.00
155	SP502	ST502	8	43	5	.36	6.92	300	38	1.29	1.00
155	SP503	ST503	8	48	6	.72	4.17	200	25	1.73	1.00
155	SP590	ST590	2	30	15	.03	9.95	300	150	.33	1.00
SUBTOTAL			56	617	11	.39	6.16	3,800	68	1.15	1.00
TOTAL			45,765	2,694,959	59	2.79	18.51	49,887,600	1,090	12.48	1.00

START 5/07/1998 16.30.18.90 I M S L O G T A P E R E P O R T S (V1M9) END 5/07/1998 17.46.38.40 PAGE 23

CPU USAGE REPORT

REGN	PROGRAM	TRANCODE	OBSERV COUNT	*******CPU TIME (MSECS)******* TOTAL	MEAN	SDEV	ELAPSE /CPU RATIO	******ELAPSE TIME (MSECS)***** TOTAL	MEAN	SDEV	TRANS /OBS	
>CTL TASK			45,765	41,913	1		109.26	4,579,500	100		11,406.75	1)
GRAND TOTAL			45,765	2,736,872	60		1.67	4,579,500	100		45,627	2)

1) AVG. NO. OF TRANSACTIONS PROCESSED PER CHECKPOINT
2) TOTAL NO. OF TRANSACTIONS PROCESSED IN REPORT PERIOD

Figure 154 (Part 2 of 5). Sample IMSPARS Reports

```
START  5/07/1998  16.30.18.90              I M S    L O G   T A P E   R E P O R T S (V1M9) END  5/07/1998 17.46.38.40 PAGE      1
                                                                CPU USAGE REPORT
                                                          ______________________________

              PROCESSING TIME              |               ELAPSE TIME              |
              _______________              |               ___________              |
                                           |                                        |
     AVERAGE  STD-DEV/AVG       MAX VALUE  |     AVERAGE  STD-DEV/AVG       MAX VALUE |
       0.198        6.082          59.497  |       3.670       21.125     4.579.500 |
                                           |                                        |
       RANGE COUNT IN     ALL REGIONS      |       RANGE COUNT IN     ALL REGIONS     |
      SC MIL   RANGE      EXCL CTL REG     |      SC MIL   RANGE      EXCL CTL REG    |
  TO MAXIMUM     490|*                     |  TO MAXIMUM     516|**                   |
       1.000     235|*                     |      10.000     311|*                    |
       0.500     224|*                     |       5.000     266|*                    |
       0.300     312|*                     |       3.000     330|*                    |
       0.200     424|*                     |       2.000     293|*                    |
       0.150     623|**                    |       1.500     258|*                    |
       0.100     672|**                    |       1.200     354|*                    |
       0.075     609|**                    |       0.900     539|**                   |
       0.050    1429|****                  |       0.600     982|***                  |
       0.025    8574|********************  |       0.300    9743|******************** |
                    ______________________ |                    ____________________ |
            --------|   |   |   |   |   |  |            --------|   |   |   |   |   | |
    TOTAL=    13,592   10  20  30  40  50% |    TOTAL=    13,592   10  20  30  40  50% |
-----------------------------------------------------------------------------------------------------------------------------------
```

Figure 154 (Part 3 of 5). Sample IMSPARS Reports

```
START  5/07/1998  16.30.18.90             I M S    L O G   T A P E   R E P O R T S (V1M9) END  5/07/1998 17.46.38.40 PAGE      1
                                                   RESOURCE AVAILABILITY

    TOTAL IMS UP TIME    1.16.20    (HH.MM.SS)
                  <-----IN USE---->    <-----IDLE------>    <--UNAVAILABLE-->                <----SUBQ6TIME---->
     PROGRAM        HH.MM.SS   PCT       HH.MM.SS   PCT       HH.MM.SS  PCT                    HH.MM.SS.T

     CFDCA04              44   1.0        1.15.35   99.0             0    .0
     CFDCC01               1    .0        1.16.19  100.0             0    .0
     CFDCC02            1.45   2.3        1.14.35   97.7             0    .0
     CFDCC05              40    .9        1.15.39   99.1             0    .0
     CFDCT03              55   1.2        1.15.24   98.8             0    .0
     CFN@AIG              26    .6        1.15.54   99.4             0    .0
     CFN@CTMS              2    .0        1.16.18  100.0             0    .0
     CFN@LA1               6    .1        1.16.13   99.9             0    .0
     CFNAA1                6    .1        1.16.14   99.9             0    .0
     CFNAA2               26    .6        1.15.54   99.4             0    .0
     CFNAA3               16    .4        1.16.03   99.6             0    .0
     CFNAC2                5    .1        1.16.15   99.9             0    .0
     CFNAE1               38    .8        1.15.41   99.2             0    .0
     CFNAG1               11    .2        1.16.08   99.8             0    .0
     CFNAG2                0    .0        1.16.19  100.0             0    .0
     CFNAG4                5    .1        1.16.14   99.9             0    .0
     CFNAH1               44   1.0        1.15.36   99.0             0    .0
     CFNAH2               30    .6        1.15.50   99.4             0    .0
     CFNAH3               11    .2        1.16.09   99.8             0    .0
     CFNAL2                2    .0        1.16.17  100.0             0    .0
     CFNAL9               10    .2        1.16.09   99.8             0    .0
     CFNAN3               20    .4        1.16.00   99.6             0    .0
     CFNAQ1               53   1.2        1.15.26   98.8             0    .0
     CFNAX3                5    .1        1.16.15   99.9             0    .0
     CFNBOA1              14    .3        1.16.06   99.7             0    .0
```

Figure 154 (Part 4 of 5). Sample IMSPARS Reports

START 5/07/1998 16.30.18.90 I M S L O G T A P E R E P O R T S (V1M9) END 5/07/1998 17.46.37.50 PAGE 1

DATA BASE UPDATE ACTIVITY

DATABASE	DDNAME	# BLOCKS UPDATED	# INSERTS GENERATED	# DELETES GENERATED	# REPLACE GENERATED	# CI SPLITS	# CA SPLITS	# DB OPEN CALLS	****FIRST UPDATE**** MM DD YY	HH SS TH	*****LAST UPDATE**** MM DD YY	HH MM SS TH
#RTDMAB1		8	66	24	3	N/A	N/A	0	05/07/98	16.54.59.5	05/07/98	16.55.06.00
#RTDMAE1		8	0	87	6	N/A	N/A	0	05/07/98	16.54.54.0	05/07/98	16.55.06.10
#RTDMAG1		9	69	26	26	N/A	N/A	0	05/07/98	16.54.59.4	05/07/98	16.56.18.80
CFDBLP		726	0	0	778	N/A	N/A	0	05/07/98	16.34.44.7	05/07/98	16.47.37.70
CFDHFP		1,115	2,586	22,254	2,169	N/A	N/A	0	05/07/98	16.30.19.0	05/07/98	16.52.04.30
		620	13	0	35,374	N/A	N/A	0	05/07/98	16.32.45.0	05/07/98	16.52.03.80
CFDHFY01		2,370?	1,272	1,578	0	N/A	N/A	0	05/07/98	16.30.19.0	05/07/98	16.52.04.30
CFDLFP		68	0	0	74	N/A	N/A	0	05/07/98	16.32.45.0	05/07/98	16.52.03.80
CFNBLP		308	0	0	313	N/A	N/A	0	05/07/98	16.30.19.3	05/07/98	16.33.44.40
CFNCNP		426	168	110	452	N/A	N/A	0	05/07/98	16.30.19.3	05/07/98	17.37.14.10
CFNCNX		111?	56	55	0	N/A	N/A	0	05/07/98	16.31.35.4	05/07/98	16.37.40.60
CFNCSP		8	0	20	22	N/A	N/A	0	05/07/98	16.31.07.4	05/07/98	17.11.42.00
CFNCSY01		2?	1	1	0	N/A	N/A	0	05/07/98	16.31.07.4	05/07/98	16.31.07.40
CFNGLP		25	407	0	0	N/A	N/A	0	05/07/98	16.33.05.3	05/07/98	17.22.14.00
CFNGLX		138?	138	0	0	N/A	N/A	0	05/07/98	16.33.05.3	05/07/98	17.22.14.00
CFNGLY01		107?	107	0	0	N/A	N/A	0	05/07/98	16.33.05.3	05/07/98	17.22.14.00
CFNHDP		51	34	0	27	N/A	N/A	0	05/07/98	16.44.28.0	05/07/98	17.02.16.90

Figure 154 (Part 5 of 5). Sample IMSPARS Reports

Appendix M. IPCS Commands for Formatting CICS Dumps

The interactive problem control system, IPCS, is a tool provided in the MVS system to aid in diagnosing software failures. IPCS provides formatting and analysis support for dumps and traces produced by MVS, other program products, and applications that run on MVS.

M.1 The General Format

IPCS commands to format a CICS dump are, in general, of the form:

```
VERBEXIT CICS520 'DEF=n,xx=nn,.....,TRS=<TRANID=tttt,KE_NUM=nnnnn>'
```

where

VERBEXIT CICS520 is required.

DEF=n Sets the default reporting level nn.

xx=n Sets the reporting level nn for some special control block.

TRS= Invokes printing of the trace table for particular tasks or transactions.

M.2 How To Set the Reporting Levels

DEF= is normally the most important parameter. It defines how much data the user receives. The default is level 0 if you omit this parameter, unless you omit all parameters, in which case it is DEF=3. The levels are:

- 0 means suppress this control block.
- 1 means print a summary only.
- 2 means print the control block in dump format.
- 3 means print the control block in both summary and dump format.

Your users get this amount of data for all control blocks, except for those specified separately.

If you specify DEF=0, you then get any data printed for those areas specified explicitly. If you specify DEF=3, you get the maximum for every area except those your user specifically exclude.

M.3 How to Specify Which Control Blocks to Print

You specify particular control blocks for which you want a different level of output, either more or less, than the default specified by DEF=n by using the codes in Table 35.

Table 35 (Page 1 of 2). CICS TS IPCS Control Block Formatting Options

Name	Value	Control Block Description
AI	0,2	Autoinstall model manager
BR	0,1,2,3	Bridge domain.
CC	0,2	CICS catalog domain
CP	0,2	CPI static storage area
CSA	0,2	CICS common system area
DD	0,1,2	Directory manager domain
DLI	0,2	CICS DL/I interface
DM	0,1,2,3	Domain manager
DS	0,1,2,3	Dispatcher domain
DU	0,2	Dump domain
FCP	0,2	File control program
ICP	0,2	Interval control program
IND	0,1,2,3	Page number indexes for the formatted output
JCP	0,2	JCA
KE	0,1,2,3	CICS kernel
LD	0,1,2,3	Loader domain
LG	0,1,2,3	Log manager domain
LM	0,1,2,3	Lock manager domain
ME	0,2	Message domain
MN	0,1,2,3	Monitoring domain
MRO	0,2	CICS multiregion operation
PA	0,2	Parameter manager domain
PCT	0,2	Program control table
PG	0,1,2,3	Program manager domain
PR	0,2	Partner resource manager (PRM) static area and table
RM	0,2	Recovery manager
SM	0,1,2,3	Storage manager domain
SSA	0,2	Static storage areas
ST	0,1,2,3	Statistics domain
SZ	0,1	Front End Programming Interface (4)
TCP	0,1,2,3	Terminal control program
TDP	0,2	Transient data program
TI	0,1,2,3	Timer domain
TMP	0,2	Table manager program
TR	0,1,2,3	Trace domain

Name	Value	Control Block Description
TRS	trace selection parameter(s)	Trace entries within SDUMP
TS	0,1,2,3	Temporary storage
UEH	0,2	User exit handler
US	0,1	User domain
XM	0,1,2,3	Transaction manager
XRF	0,2	Extended recovery facility

Table 35 (Page 2 of 2). CICS TS IPCS Control Block Formatting Options

Your users can get only what was dumped originally. CICS normally dumps the areas shown in Table 36 on page 329

M.4 How to Specify Trace Data to Print

Users can print out parts of the trace table, using the TRS= parameter. The following is a summary of what option parameters you can specify in the formatting report utility DFHTUR520:

- **PAGESIZE=(value)**. Specifies the number of lines printed on a page. You can specify a value in the range 20 through 9999 lines per page. If you specify an incorrect value, CICS issues an error message and stops the trace. The default value is 55.

  Note: This parameter is not valid for printing GTF trace entries

- **ABBREV|SHORT|FULL**. Specifies how much of each trace entry you want printed. If you specify this statement, it must always be the first statement in either the PARM parameter or the DFHAXPRM data set.

  – **ABBREV**. Indicates that you require the abbreviated, one-line-per-entry, form of trace print.

  – **SHORT**. Indicates that you require the short formatted print of the data in each entry. This consists of the information in the abbreviated format entry, and the following elements from the interpretation string of the fully formatted entry:

    - Interpreted parameter list
    - Return address
    - Time
    - Interval

  – **FULL**. Indicates that you want a fully formatted print of all the data in each entry. This is the default.

- **ALL**. Specifies that all trace entries in the auxiliary trace data set are to be printed. This is the default.

- **ENTRY_NUM=({nnnnnn|nnnnnn-nnnnnn},{nnnnnn|nnnnnn},.,.,.)** . Specifies the sequence numbers of one or more trace entries that you want to print. Each sequence number can be up to six digits long. If you specify a range of sequence numbers by using xxxxxx-yyyyyy, the second sequence number (yyyyyy) must be larger than the first (xxxxxx).

- **EXCEPTION**. Specifies that only exception trace entries in the auxiliary trace data set are to be printed.

  Note: This parameter is not valid for printing GTF trace entries.

- **INTERVAL={00.128|number of seconds}**. Specifies the interval between auxiliary trace entries after which entries are highlighted with an asterisk, as follows:

  – In abbreviated trace format, the asterisk appears to the left of the sequence number.

  – In full trace format, the asterisk appears as the next character after the printed time interval (as it does in releases prior to CICS Transaction Server for OS/390 Release 2 where a system-imposed time interval of 0.0128 second applies).

  If successive auxiliary trace entries are written at intervals equal to or greater than this limit, they are highlighted in the same manner.

If successive auxiliary trace entries are written at intervals smaller than this limit, they are not highlighted. They are, however, written, formatted and printed.

If you specify no INTERVAL value, a default of 0.128 second applies.

Users can specify interval values in the range zero seconds (where all trace entries would be highlighted) through 99.9999999999 seconds.

Note: The interval extends to ten decimal places. Zeros are padded from the right.

- **KE_NUM=(xxxx,xxxx,.,.)**. Specifies that only the entries for tasks with the specified hexadecimal kernel task numbers are printed.
- **PAGESIZE=(value)**. Specifies the number of lines printed on a page. You can specify a value in the range 20 through 9999 lines per page. If you specify an incorrect value, CICS issues an error message and stops the trace. The default value is 55.

  Note: This parameter is not valid for printing GTF trace entries.
- **TASKID=({id|id-id},,{id|id-id},.,..)**. Specifies the task identifiers (ID) of one or more tasks for which trace entries are to be printed. An ID value can be in any of the following forms, to compare with the task field in the formatted trace data:
  - Any number up to five decimal digits long.
  - Any of the character strings JAS, J01 through J99, III, TCP, or DSTCB
  - Any nonnumeric two-character domain ID of the attaching domain (for non-TCA) tasks.

  Users can specify a range of task identifiers of the five decimal digit form by using a hyphen (for example, TASKID=nnnnn-nnnnn).
- **TERMID=(tttt,tttt,.,.,.)**. Specifies the terminal identifiers (tttt) of one or more terminals for which trace entries are to be printed.

  If users use the TERMID parameter to specify the trace entries you want formatted, the DFHTU520 program selects all the trace entries that are associated with any transaction-attach trace entries it finds containing the terminal identifiers you specify. For more information about how trace entries for tasks are associated with transaction-attach trace entries, see *OS/390 IPCS User Guide, GC28-1756*.
- **TRANID=(tttt,tttt,.,.,.)**. Specifies the transaction identifiers of one or more transactions for which trace entries are to be printed.

  If you use the TRANID parameter to specify the trace entries your user wants formatted, the DFHTU520 program selects all the trace entries that are associated with any transaction-attach trace entries it finds that contain the transaction identifiers you specify.
- **TIMERG=(hhmmss-hhmmss,hhmmss-hhmmss,.,.,.)**. Specifies the time period or periods for which trace entries are to be printed. Time periods are shown by pairs of values represented as hours (hh), minutes (mm), and seconds (ss) separated by a hyphen. The ending value of each pair must be later than the starting value.

  The DFHTU520 program converts the store-clock (STCK) values in the trace entries to whole seconds prior to comparing against the time range you specify. Fractions of a second are ignored; that is, all times are rounded down to the nearest whole second, which means in effect that the minimum time span can be up to 2 seconds. For example, if you specify TIMERG=153000-153001, the DFHTU520 program includes all trace entries with times of 153000.00000000 to 153001.99999999 inclusive.

Note: This parameter is not valid for printing GTF trace entries.

- **TYPETR=({ddxxxx|ddxxxx-xxxx},{ddxxxx|ddxxxx-xxxx})**. Specifies the trace entry identifiers for the particular domain entries, specified by the domain ID and a point ID within the domain.

DD	Represents the domain identifier:
AP	Application
DD	Directory manager
DE	DCE services
DM	Domain manager
DS	Dispatcher
DU	Dump
EX	External CICS interface
GC	Global catalog
KE	Kernel
LC	Local catalog
LD	Loader
LG	Log manager
LM	Lock manager
ME	Message
MN	Monitoring
NQ	Enqueue
PA	Parameter manager
PG	Program manager
RM	Recovery manager
SM	Storage manager
ST	Statistics
TI	Timer
TR	Trace
US	User
XM	Transaction manager
XS	Security manager

xxxx Represents the point ID within the domain in the form of a four-character hexadecimal value (0000-FFFF). You can specify a range of point IDs by using a hyphen.

These are also the parameters you supply to the trace print utility DFHTU520 to control the printing of the trace. They are supplied in that case through the DFHAXPRM DD * statement.

M.5 SDUMP Parameters

When the SDUMP is invoked, the options shown in Table 36 must have been included as a minimum; otherwise, IPCS will be unable to find many of the control blocks in CICS even if they are present in the dump. CEMT P DUMP will always include these.

Table 36. CICS TS IPCS Dump Options

Options	Description
ALLPSA	Prefix storage area for all processors
CSA	Common storage area.
GRSQ	Global resource serialization queues
LPA	Link pack area
NUC	Non-page-protected areas of the DAT-on nucleus
RGN	Entire private area of the CICS address space
SQA	System queue area
SUMDUMP	The summary dump function
TRT	GTF, system trace, and master trace data

Appendix N. Special Notices

This publication is intended to help technical professionals understand the differences between CICS VSE Local DL/I and MVS/ESA CICS Transaction Server with IMS/ESA DBCTL. It also explains how to migrate application systems from VSE to OS/390. The information in this publication is not intended as the specification of any programming interfaces that are provided by IMS/ESA Version 6 and CICS Transaction Server for OS/390 Version 1. See the PUBLICATIONS section of the IBM Programming Announcement for IMS/ESA Version 6 Release 1 and CICS Transaction Server for OS/390 Version 1 Release 2 for information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM®	AFP
AnyNet®	AS/400®
AT®	BookManager®
C/370	CICS®
CICS/ESA®	CICS/MVS®
CICS/VSE®	DB2®
DFSMS	DFSORT
Enterprise Systems Architecture/390®	ESA/390
ESCON®	GDDM®
Hiperbatch	IBM®
IMS	IMS/ESA®
OpenEdition®	OS/2®
OS/390	Parallel Sysplex
PR/SM	PROFS®
RACF	RMF
RS/6000	S/390®
SOMobjects	SP
SQL/DS	System/370
System/390®	VM/ESA®
VSE/ESA	VTAM®
400®	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix O. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

O.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 335.

- *IMS/ESA V5 Performance Guide, SG24-4637*
- *CICS Transaction Server for OS/390 V1R2 Implementation Guide, SG24-2234*
- *VSE to OS/390 Migration Workbook, SG24-2043*

O.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

O.3 Other Publications

These publications are also relevant as further information sources:

- *IMS/ESA, Installation Volume I, GC26-8736*
- *IMS/ESA, Installation Volume II, GC26-8737*
- *IMS/ESA, DBRC Guide and Reference, SC26-8733*
- *IMS/ESA, Administration Guide: System, SC26-8730*
- *IMS/ESA, Operations Guide, SC26-8741*
- *IMS/ESA, Diagnosis Guide and Reference, LY37-3731*
- *IMS/ESA, Utilities Reference: Database Manager, SC26-8769*
- *IMS/ESA, Performance Analyzer User Guide, SC26-9088*
- *IMS System Utilities/Data Base Tools, High Speed Sequential Retrieval Guide, SH21-0548*
- *IMS Database Control Suite User Guide, SC26-9201*
- *CICS Transaction Server for OS/390 Installation Guide, GC33-1681*
- *CICS Transaction Server for OS/390 CICS RACF Security Guide, SC33-1701*

- *CICS Transaction Server for OS/390 CICS IMS Database Control Guide, SC33-1700*
- *CICS Transaction Server for OS/390 CICS Customization Guide, SC33-1683*
- *CICS Transaction Server for OS/390 CICS Shared Data Table Guide, SC33-1702*
- *CICS Transaction Server for OS/390 CICS Operation and Utility Guide, SC33-1685*
- *CICS Transaction Server for OS/390 CICS Performance Guide, SC33-1699*
- *CICS Transaction Server for OS/390 CICS Resource Definition Guide, SC33-1684*
- *OS/390 RACF Command Reference, SC28-1919*
- *OS/390 MVS System Codes, GC28-1780*
- *OS/390 MVS System Commands, GC28-1781*
- *OS/390 MVS JCL Reference, GC28-1757*
- *OS/390 MVS Authorized Assembler Services Guide, GC28-1763*
- *OS/390 JES2 Macros, SC28-1795*
- *OS/390 IPCS User Guide, GC28-1756*
- *CICS for VSE/ESA 2.3 System Control Statement, SC33-6613*
- *VSE Messages and Codes, Volume 1, SC33-6507*
- *VSE/POWER Application Programming Guide, SC33-6574*
- *CICS for VSE/ESA 2.3 Resource Definition (Macro), SC33-0709*
- *CICS for VSE/ESA 2.3 System Programming Reference, SC33-0711*
- *VSE to MVS Differences and Migration Guide, GG66-3113*

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** — to order hardcopies in the United States

- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**

- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

abend. Abnormal end of task

access method. A technique for moving data between main storage and input/output devices, for example, VSAM, VTAM.

access method control block (ACB). A control block that links an application program (for example, a CICS system) to an access method (for example VSAM or VTAM). In communication with DL/I, an ACB is used only when the underlying access method is VSAM.

access method services (AMS). A utility program for the definition and management of VSAM data sets.

active libraries. The libraries from which IMS draws its execution information when online change is used.

address space. A range of up to two gigabytes of contiguous virtual storage addresses that the system creates for the user. Unlike a data space, an address space contains user data and programs, as well as system data and programs, some of which are common to all address spaces. Instructions execute in an address space, not a data space.

addressing mode (AMODE). (1) The mode, 24-bit or 31-bit, in which a program holds and processes addresses. The AMODE linkage-editor control statement specifies the addressing mode of the load module produced. (2) A program attribute that refers to the address length that a program is prepared to handle on entry. Addresses may be either 24 bits or 31 bits in length. In 24-bit addressing mode, the processor treats all virtual addresses as 24-bit values: in 31-bit addressing mode, the processor treats all virtual addresses as 31-bit values. Programs with addressing mode of ANY can receive control in either 24-bit or 31-bit addressing mode. (3) A control statement that defines the addressing mode of the load module produced by the linkage editor.

application control blocks (ACB). The control blocks created from the output of DBDGEN and PSBGEN and placed in the ACB library for use during online and DBB region type execution of IMS.

application identifier (VTAM applid). The name by which a logical unit known in a VTAM network. The CICS applid is specified in the APPLID system initialization parameter.

Application Migration Aid. A program that simplifies conversion of assembler language and COBOL applications from macros to command-level. The Application Migration Aid reads assembler language and COBOL source code and writes a new source file, converting the simpler macros to equivalent API commands and providing guidance on the complex macros.

archiving logs. The process of copying records or logs of IMS activity from the online log data set, which is temporarily recorded on DASD, to the system log data set, which is stored on DASD, tape, or mass storage.

archive process. The process of archiving the IMS/ESA OLDS to a system log data set (SLDS) on either tape or DASD.

autoinstall. A method of creating and installing terminal definitions (TCT entries) dynamically as terminals log on, and deleting them at logoff.

autoinstall control program. A user-replaceable CICS program used to select some of the data needed to automatically install terminals, notably the CICS terminal identifier (TERMID) and the model name to be used in each instance.

batch checkpoint/restart.. The facility that enables batch processing programs to synchronize checkpoints and to be restarted at a user-specified checkpoint.

batch image copy. A copy of a database or area that reflects the state the data at a moment in time when no updates were being made. The Database Image Copy utility (DFSUDMP0) produces batch image copies, which IMS utilities use when recovering from failures.

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

batch processing. (1) Type of data processing in which a number of input items are grouped for processing serially with a minimum of operator intervention and no end-user interaction. (2) Serial processing of computer programs. (3) Pertaining to the technique of executing a set of computer programs so that each is completed before the next program of the set is started.

batch processing program. An application program that has access to databases and MVS data management facilities but does not have access to the IMS control region or its message queues.

CDBC. (1) Name of a CICS transaction that connects CICS to DBCTL. CDBC can also disconnect CICS from DBCTL. (2) CICS transient data destination used as a data log for DBCTL DFHDB81xx messages. Other

DBCTL messages use either the terminal or the console.

CDBI transaction. CICS transaction that inquires about the status of the interface between CICS and DBCTL.

CDBM transaction. A CICS-supplied transaction from which commands can be issued to IMS DBCTL. With CICS for MVS 4.1, and IMS/ESA 5.1 or later, you can use CDBM to issue most of the IMS operator commands that are valid for DBCTL across the DRA interface to DBCTL to display and change the state of selected resources.

checkpoint. In IMS, the point at which an application program commits that the changes it has made to a database are consistent and complete, and releases database segments for use by other programs. You can request checkpoints at appropriate points in a program to provide places from which you can restart that program if it, or the system, fails.

For an IMS system, a point in time from which the system can start again if a failure makes recovery necessary. The checkpoint is performed by IMS itself.

cold start. The starting of IMS when it is initialized for the first time or or when some error condition prevents a warm or emergency restart.

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

database control (DBCTL). An environment allowing full-function database and DEDBs to be accessed from one or more transaction management subsystems.

database data set (DBDS). A data set containing all or part of a database.

database description (DBD). The collection of macro parameter statements that define the characteristics of a database, such as the database′s organization and access method, the segments and fields in a database record, and the relationship between types of segments.

database description generation (DBDGEN). The process by which a DBD is created.

database recovery. The process of restoring a physically damaged DBDS by merging an image copy and logs or change accumulations.

Database Recovery Control (DBRC). A feature of the IMS Database Manager that facilitates easier recovery of IMS databases. DBRC maintains information required for database recoveries, generates recovery control statements, verifies recovery input, maintains a separate change log for database data sets, and supports sharing of IMS databases and areas by multiple IMS subsystems.

database reorganization. The process of unloading and reloading a database to optimize physical segment adjacency or to modify the DBD.

Data Language/I (DL/I). The IMS data manipulation language, a common high-level interface between a user application and IMS. DL/I calls are invoked from application programs written in languages such as PL/I, COBOL, VS Pascal, C, and Ada. It can also be invoked from assembler language application programs by subroutine calls. IMS lets the user define data structures, relate structures to the application, load structures, and reorganize structures.

DL/I address space. An address space used by the online IMS control program to contain most of the DL/I code and control blocks. This option can be selected for the online IMS environment to provide an alternative virtual storage configuration.

dynamic allocation/deallocation. A function that removes the requirement to allocate IMS databases, area data sets, and certain system data sets through job control language. A data set is allocated during IMS initialization or when it is first used and is deallocated when it is no longer used (that is, closed or stopped).

Fast Path. IMS functions for applications that require good response characteristics and that may have large transaction volumes. Programs have rapid access to main-storage databases (to the field level), and to direct-access data entry databases. Message processing is grouped for load balancing and synchronized for database integrity and recovery.

generalized sequential access method (GSAM). A database access method providing accessing support for simple physical sequential data sets, such as tape files, SYSIN, SYSOUT, and other files that are not hierarchic in nature. Available only in BMP and Batch.

GRPLIST. System initialization parameter that specifies the name (1 to 8 characters) of a list of groups on the CSD. The resource definitions in all the groups in the specified list are loaded during initialization when CICS performs a cold start.

hierarchy. In a database, a tree of segments beginning with the root an proceeding downward to dependent segment types. No segment type can be dependent on more than one other segment type.

IMS subsystem. An individual batch or online IMS control program executing in an MVS address space.

IMS system log. Logically, a single log made up of on-line data sets (OLDSs) and write-ahead data sets (WADSs).

IMSPARS. A Program Offering product that analyzes the IMS/ESA log file.

input/output program communication block (I/O PCB). A DC-PCB provided automatically by IMS to an application program that executes in a communication system with the DC feature. The I/O PCB is the mechanism which a program obtains an input message from a terminal and returns a reply to the terminal that originated the input message.

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

MLPA. Modifiable Link Pack Area. A part of MVS storage.

monitoring. Running a hardware or software tool to measure the performance characteristics of a system.

online. Applicable in the IMS DB/DC, DBCTL, and DCCTL environments, unless otherwise indicated.

online change. The process of adding, deleting, or replacing various IM resources without stopping the system to redefine them.

online forward recovery. In a remote site recovery environment, the process by which a stopped shadow database or area is brought to currency by the tracking subsystem with the database or area on the active subsystem.

online image copy. The process of creating an image copy while the DBDS is online. Also, the image copy created by the process.

OLDS. The IMS/ESA online log data set that contains the log records of all IMS/ESA activity.

program specification block (PSB). The control block that describes databases and logical message destinations used by an application program. A PSB consists of one or more PCBs.

program specification block generation (PSBGEN). The process by which a PSB is created.

PUNCH. The PUNCH command (a VSE command) causes the contents of one or more library members to be *punched* to the output device SYSPCH.

Note: *Punched* indicates the record format of the data produced, not the type of storage medium used for output.

recovery control (RECON) data sets.. Data sets in which Data Base Recovery Control stores information about logging activity and events that might affect the recovery of databases.

resource definition online (RDO). The recommended method of defining resources to CICS. Resource definitions are created interactively with the CEDA transaction, or by using the utility DFHCSDUP. Both methods store definition in the CICS system definition data set (CSD). At CICS initialization, CSD definitions are selectively installed as CICS system tables, controlled by a user-supplied list of definitions. CEDA-defined resource definitions can be installed while CICS is active and used immediately.

RMODE. In MVS and VSE, an attribute that specifies the residence mode of the load module produced by the linkage editor. The possible values are RMODE(24) and RMODE(ANY). A program link-edited with RMODE(24) must reside below the 16MB line. If a program is link-edited with RMODE(ANY) CICS loads it above the 16MB line if possible.

simple IMS/ESA checkpoint. A term used to denote when the IMS/ESA system control blocks are written to the system log.

system log data set (SLDS). The system log data set, which is normally the tape data set that contains the archived OLDS.

stage 1 system definition. The first part of the process of defining an IMS system. Stage 1 checks input specifications and generates a series MVS job steps that are the input to stage 2.

stage 2 system definition. The second part of the process of defining an IMS system. Stage 2 builds IMS system libraries, execution procedures, and the IMS online control program tailored to support the desired set of IMS functions. Stage 2 then stores these in an IMS library.

staging libraries. Those libraries that are modified by offline function in a system using online change. Changes are first applied to the staging libraries, which are then copied to the inactive libraries.

sync point. A term used in IMS/ESA to denote the time when any committed records not previously written to the database are written to the WADS and database.

transaction. A message from a terminal or an application program that causes the application program logic to be executed.

unit of work. A distinct unit of processing that can be released by the application program for use by other programs in the system.

write ahead data set (WADS). IMS/ESA write ahead data set, which holds the committed log records before writing them to the OLDS.

List of Abbreviations

ACB	Access control block in VSAM. Application control block in IMS.
ADS	(Read or write) ahead data set.
AID	Attention identifier.
AMS	Access method services.
AO	Automatic operation.
APA	All points addressable.
API	Application programming interface.
APF	Authorized program facility.
APPC	Advanced program-to-program communication; also advanced peer-to-peer communication.
APPLID	VTAM application ID.
BMP	Batch message processing (IMS region).
CICS	Customer information control system.
CPU	Central processing unit.
CSA	Common system area for CICS, common storage area for MVS.
CSD	CICS system definition.
DASD	Direct access storage device.
DBCTL	Database control.
DBD	Database description.
DBDGEN	Database description generation.
DBDS	Database data set
DBRC	IMS/VS Database recovery control.
DBDS	Database data set.
DBT	Database tool.
DB2	Database 2.
DCB	Data set control block.
DCT	Destination control table.
DD	Data set definition.
DDIR	Database directory.
DDL	Data definition language.
DEDB	Data entry database.
DL/I	Data language I.
DLS	DL/I separate address space.
DLISAS	DL/I separate address space.
DRA	Database resource adapter
DSA	Dynamic storage area.
DTB	Dynamic transaction backout.
ESA	Enterprise systems architecture.
ESDS	Entry sequenced data set.
FBA	Fixed block architecture.
FCT	File control table.
GN	Get next.
GSAM	Generalized sequential access method.
GU	Get unique.
HD	Hierarchic direct.
HDAM	Hierarchic direct access method.
HIDAM	Hierarchic indexed direct access method.
HISAM	Hierarchic indexed sequential access method.
HSSP	High-speed sequential processing.
HSSR	High-speed sequential retrieval.
I/O PCB	Input/output program communication block, also called TPPCB.
IBM	International Business Machines Corporation.
ICBC	Industry and Commercial Bank of China.
IMF	International Monetary Fund.
IMS	Information management system.
IMSPARS	IMS/ESA performance analysis and reporting system.
IPCS	Interactive problem control system.
IPL	Initial program load.
ISC	Intersystem communication.
ISPF	Interactive system productivity facility.
ITSO	International Technical Support Organization.
JCL	Job control language.
JES	Job entry system (component of operating system JES2 and JES3).
KSDS	Key sequenced data set.
LLA	Library lookaside.
MCT	Monitor control table.
MLPA	Modified link pack area (MVS).
MVS	Multiple virtual system.
OEM	Original equipment manufacturer.
OLDS	Online log data set.

OSAM Overflow sequential access method.

PARMLIB MVS parameter library data set.

PCB Program communication block; can be TPPCB (I/O PCB) or DBPCB (database).

PDIR Program directory.

PDS Partition data set.

PLT Program load table.

PR Performance Reporter for OS/390.

PROCLIB Procedure library data set; IMS procedures are in IMSESA.PROCLIB and MVS procedures are in SYS1.PROCLIB.

PROFS Professional Office System.

PSB Program specification block, made up of PCBs, both I/O and database.

RACF Resource access control facility.

RDO Resource definition online.

RECON Recovery control.

RSL Resource security level.

RSR Remote site recovery.

SAFE II System for advanced financial environment interactive interface.

SIT CICS system initialization table.

SLDS System log data set.

SMP System modification program.

SPA Scratch pad area.

SRT System recovery table (CICS).

SSA Segment search argument.

SVA System virtual area.

TCB Task control block.

TCT Terminal control table.

TCTTE Terminal control table terminal entry.

TLT Terminal list table.

TRA Trace anchor

TSO Time sharing option.

UNDP United Nations development programs.

UOW Unit of work.

UPSI User program switch indicators.

VSAM Virtual sequential access method.

VTAM Virtual telecommunications access method.

VSE Virtual system extension.

WADS Write ahead data set.

WLM Work load manager.

WWW World wide web.

XRF Extended recovery facility.

Y2000 Year 2000

Index

F

G

I

J

L

M

ITSO Redbook Evaluation

Interoperability Between VSE DL/I and OS/390 IMS DBCTL
SG24-5249-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer   __Business Partner   __Solution Developer   __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

____

____

____

____

What other redbooks would you like to see published?

____

____

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

____

____

____

____

____

Interoperability Between VSE DL/I and OS/390 IMS DBCTL

SG24-5249-00

SG24-5249-00
Printed in the U.S.A.