Series 90 to Model 8 Migration

OS/3

User Guide

Environment: System 80

H

SPERRY

UP-9743

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry representative.

Sperry reserves the right to modify or revise the content of this document. No contractual obligation by Sperry regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry.

FASTRAND, ✦SPERRY, SPERRY, SPERRY✦UNIVAC, SPERRY UNIVAC, UNISCOPE, UNISERVO, UNIVAC, and ✦ are registered trademarks of the Sperry Corporation. ESCORT, MAPPER, PAGEWRITER, PIXIE, SPERRYLINK, and UNIS are additional trademarks of the Sperry Corporation.

PAGE STATUS SUMMARY

ISSUE: UP-9743
RELEASE LEVEL: 8.2 Forward

Part/Section	Page Number	Update Level
Cover/Disclaimer		
PSS	1	
Preface	1, 2	
Contents	1 thru 4	
1	1 thru 13	
2	1 thru 11	
3	1 thru 14	
4	1 thru 3	
5	1 thru 9	
6	1 thru 5	
7	1 thru 4	
Index	1 thru 3	
User Comment Sheet		

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

All the technical changes are denoted by an arrow (⟹) in the margin. A downward pointing arrow (⇓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (⇑) is found. A horizontal arrow (⟹) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is one of a series designed to instruct and guide the programmer in the use of the SPERRY Operating System/3 (OS/3). This manual specifically describes the process of migrating from the SPERRY Series 90 Data Processing System to the SPERRY System 80 model 8 Data Processing System. It is intended for Series 90 users of OS/3 releases 6.1.2 forward who are migrating to OS/3 release 8.2 on the model 8.

This manual has seven sections:

■ Section 1. Introduction to Migration

Introduces the process of migrating from Series 90 to the model 8. This section also provides information on how to approach the migration process, a suggested sequence for accomplishing the phases of migration, and considerations for the migration process.

■ Section 2. Data File Migration

Describes the processes of transcribing disk data files to the model 8. This section also discusses the data management environment of the model 8 and describes the reformatting of disk data files into MIRAM format.

■ Section 3. Source Program Migration

Describes the migration of Series 90 programs written in the following languages to the model 8:

COBOL

FORTRAN IV

RPG II

BASIC

Basic Assembly Language

ESCORT Programming Language

This section describes how to convert programs that run under basic data management (BDM) to run under consolidated data management (CDM).

- Section 4. System Generation

  Discusses considerations for generating a supervisor on the model 8. This section lists new or modified SYSGEN parameters and new system files that the Series 90 user may be unfamiliar with.

- Section 5. ICAM Migration

  Describes the migration of Series 90 user programs to the model 8. This section discusses new ICAM features and the changes a user must make to Series 90 communications physical interface (CPI) programs to run on the model 8.

- Section 6. IMS Migration

  Describes the process of converting Series 90 single-thread IMS systems to multithread. This section also discusses the conversion of IMS systems that use BDM to use CDM.

- Section 7. Software Products

  Briefly describes the migration of the following products to the model 8:

  - Screen formats
  - BEM EDT procedures
  - SORT/MERGE and SORT3 job control streams
  - Data base management system 90 (DMS 90)

Contents

2. DATA FILE MIGRATION

3. SOURCE PROGRAM MIGRATION

4. SYSTEM GENERATION

5. ICAM MIGRATION

6. IMS MIGRATION

7. SOFTWARE PRODUCTS

FIGURES

TABLES

1. Introduction to Migration

1.1. GENERAL

Your Series 90 data processing system represents a significant investment in hardware, programs, data files, and training of personnel. But a growing business makes ever increasing demands on a computer system. As a result, you may want to upgrade your present computing system to one that has more processing power, is more economical, or that meets changing needs. To help users of the SPERRY 90/25, 90/30, 90/30B, and 90/40 systems satisfy these requirements, we provide a powerful and yet inexpensive migration path for all Series 90 users – the System 80 model 8.

The model 8 offers increased processing power, new high-capacity peripheral devices, and many easy-to-use interactive software facilities. These features make the model 8 an attractive upgrade path for all Series 90 users. When you migrate to the model 8, however, you'll also be surprised at how easy the transition is. This is because the model 8 provides extensive software and hardware compatibility with your current Series 90 system.

The model 8 operates under the control of an enhanced version of the OS/3 operating system software, the same operating system you use with your Series 90 system. OS/3 supports common programming languages, program products, and data storage methods that enable you to quickly move your applications to the model 8. You can also use the same communications facilities on the model 8, so you don't have to redesign your online programs. And, many of your Series 90 peripheral devices, such as disk and tape units, are also compatible between systems. In many cases, you can simply move your programs and data files directly to the model 8 and start production immediately!

This manual provides the information you need to migrate to the model 8 quickly and efficiently. We'll begin by giving you an overview of the migration process. We'll tell you about the various migration paths you can choose from when moving to your new system, and describe the many compatibility features the model 8 shares with your Series 90 system. Later in this section, we'll provide guidelines that should help you get started in a successful migration. The rest of this manual gives the details you need to actually carry out the migration.

1.2. EASE OF MIGRATION FLOWCHART

The flowchart in Figure 1–1 gives an overview of the migration process. We've made one assumption in this flowchart: that you'll carry your programs and data files to the model 8 on your Series 90 disk devices. Since most of your Series 90 disks can be used on the model 8, this is the quickest and easiest way to migrate to your new system. Once release 8.2 is installed, you can simply mount your Series 90 disks on the model 8 and begin generating your supervisors, ICAM networks, and IMS configurations immediately. A complete list of devices that are compatible between systems is shown in 1.5.2.

However, if you decide to use the model 8 8470 disk device exclusively, you must copy your programs and data files to a temporary media such as tape or diskette on your Series 90 system. Then, you can load your programs and data files from the temporary media onto an 8470 disk. We've put the sections that deal with program and data file migration in the beginning of this manual in case you decide to migrate in this way. Section 2 tells you how to move your data files to the model 8; program migration is discussed in Section 3.

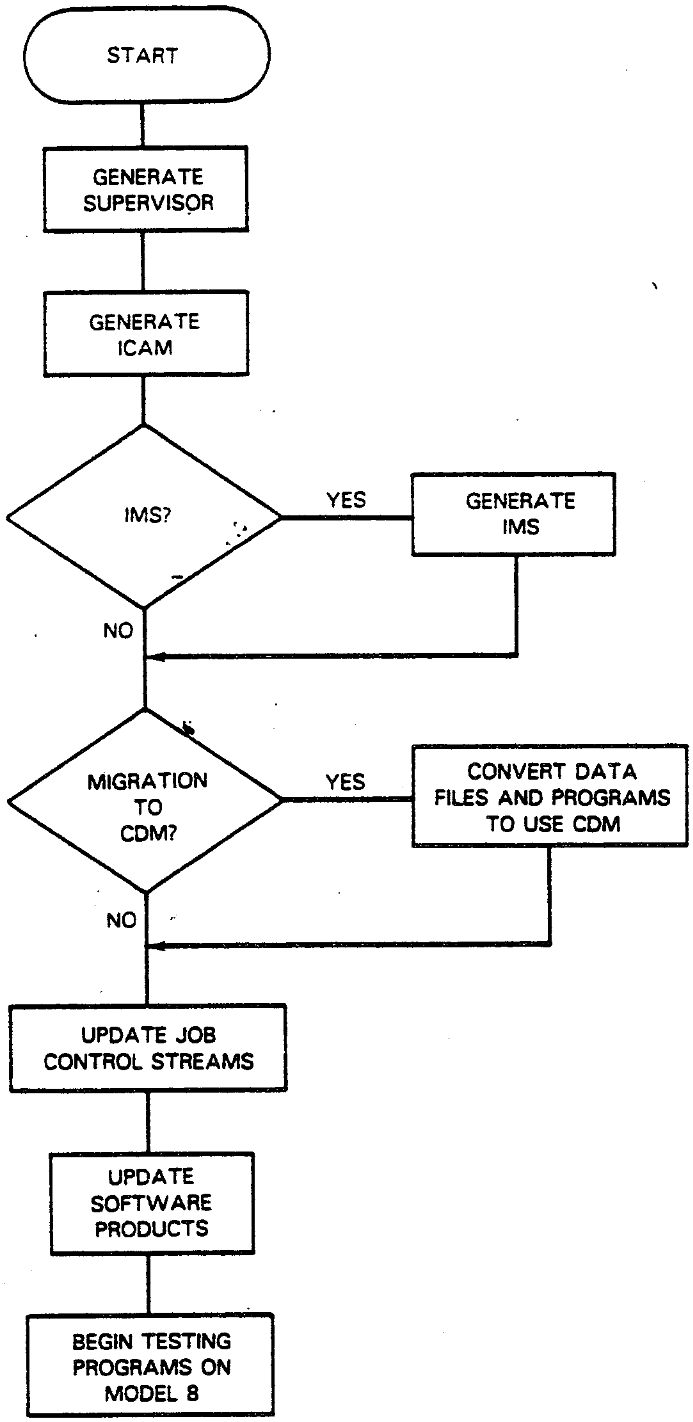
Of course, your operational needs determine the exact plan of your migration; this flowchart is simply a suggested migration plan. We think it should give you a general idea of what you must consider and point you in the right direction for a successful migration.

1.3. MODEL 8 ENVIRONMENT

Most Series 90 users operate under basic data management (BDM). BDM uses the define the file (DTF) interface and supports each of the following file access methods:

MIRAM	(multiple indexed random access method)
IRAM	(indexed random access method)
SAT	(system access technique)
SAM	(sequential access method)
DAM	(direct access method)
ISAM	(indexed sequential access method)
ASAM	(alternate sequential access method)

The model 8, however, is designed to operate primarily under consolidated data management (CDM). CDM is an enhanced version of data management that uses a common interface, the common data interface (CDI), for all files. CDI supports just one file access method, MIRAM (and IRAM, a subset of MIRAM), which offers all the processing capabilities previously provided on Series 90 by sequential, relative, direct, and indexed file access methods.



Generation includes SUPGEN and I/OGEN phases of SYSGEN. Select data management mode

DMGTMODE=$\left\{\begin{array}{l}\text{MIXED}\\ \text{CDI}\end{array}\right\}$

Update network definitions, recompile CUPs, and generate dedicated or global network.

Recompile action programs. If using new model 8 features, convert single-thread IMS systems to multithread and convert applications that use BDM to use CDM.

If not already using CDM, convert data files to MIRAM and recompile source programs to use CDM.

Modify job control streams to specify new model 8 devices.

Update screen formats (if created on release 7.1) and BEM EDT procedures.

Use sample data files; run side by side with Series 90 system.

Figure 1–1. Model 8 Migration Flowchart

To ease your migration, the model 8 supports applications that use BDM as well as those that use CDM. This lets you run your current Series 90 programs on the model 8 without changing those programs or the data files they use. On the model 8, you choose your data management environment during system generation (SYSGEN) by specifying one of two values for the DMGTMODE parameter:

```
DMGTMODE={MIXED}
         {CDI  }
```

If you specify DMGTMODE=CDI, the model 8 supports CDM only; if you accept the default DMGTMODE=MIXED, the model 8 supports BDM and CDM. When you operate in mixed mode, the model 8 is fully compatible with your Series 90 system at the load code level. A program that runs on Series 90 using BDM can run on the model 8 without change; a Series 90 program that uses CDM can also run on the model 8 without change.

We recommend that, whenever possible, you operate in a CDM-only environment. The model 8 has many advanced interactive facilities that use CDM exclusively. These facilities include screen format services (SFS), ESCORT programming language, dialog processor, the OS/3 general editor (EDT), and workstation support. While your existing applications can run on the model 8 without change, if you want to incorporate these new features into your applications, you must migrate from BDM to CDM. Basically, this involves converting your disk data files to MIRAM format and, in some cases, recompiling your programs. Here are some of the advantages MIRAM provides:

- A single access method for all file structures
- Improved file access time
- Multikey access
- Record deletion
- Higher degree of disk file sharing
- Distributed data processing (DDP) remote file sharing

Throughout this manual we'll tell you how to convert various software features to CDM. Remember that we offer mixed mode support primarily so you can continue using BDM applications after you migrate to the model 8. Whenever possible, you should convert your programs and data files to run in a CDM-only environment.

1.4. MIGRATION PATHS

There are three migration paths you can choose from when moving to the model 8. Each path is designed so that you can make the transition from BDM to CDM with a minimum of effort.

Series 90		Model 8
CDM	to	CDM
BDM	to	BDM
BDM	to	CDM

- CDM to CDM

  If you have already implemented CDM-oriented applications on your Series 90 system, you can move those applications to the model 8 without modification. Since many Series 90 disk units are supported on the model 8, your migration may simply involve carrying your Series 90 disks forward to the model 8. Figure 1–2 illustrates this process. If you want to use model 8 devices exclusively, however, you must transcribe all your data and library files to tape or diskette on your Series 90 system and then load them onto a model 8 device. We'll tell you how to do this in Sections 2 and 3.

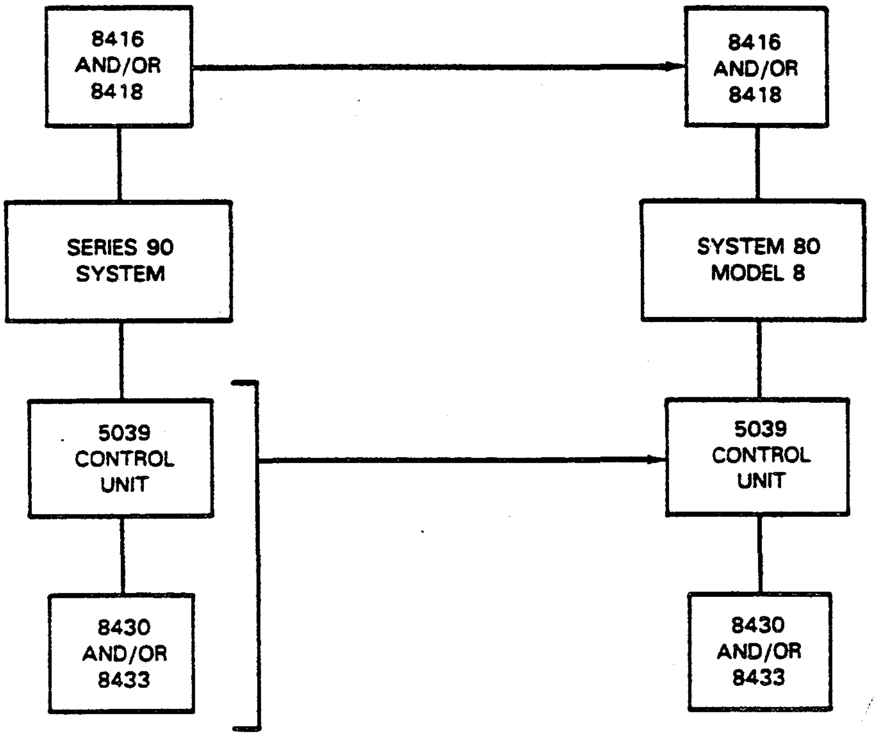


Figure 1–2. Moving Series 90 Disks to the Model 8

- BDM to BDM

The model 8 provides support for BDM applications to be run side by side with CDM applications. This mixed mode data management support is illustrated in Figure 1–3; it enables you to move your BDM applications intact to the model 8.

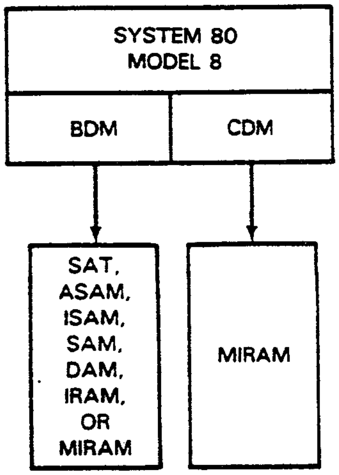


Figure 1–3. Model 8 Mixed Mode Data Management Support

Once your applications are running on the model 8 under BDM, you can make the transition to CDM at your convenience. This is a 2-step process that involves recompiling your programs and transcribing your disk data files to MIRAM format. Though you can perform these tasks at your own pace, you must be careful to coordinate the effort to ensure that when you recompile a program, you also convert the data files it uses to MIRAM. Figure 1–4 illustrates this process.

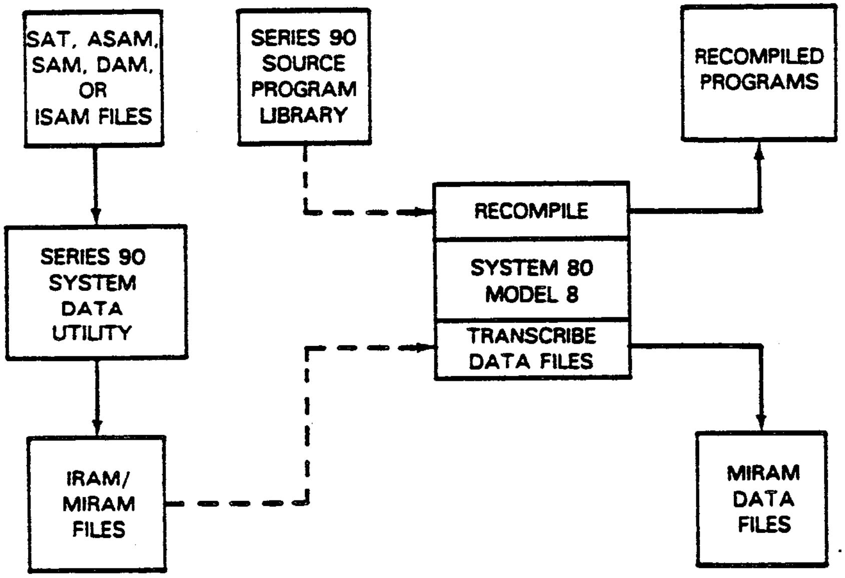


Figure 1–4. Moving Programs and Data Files to the Model 8

- BDM to CDM

You can migrate to the model 8 and change from BDM to CDM at the same time. If you make the transition to CDM on your Series 90 system, you can move your disk units to your new system as we described previously. If you make the transition to CDM on the model 8, you proceed as described in the BDM to BDM section. Regardless of which system you use, the procedure remains the same. You must recompile your programs and transcribe your disk data files to MIRAM format.

1.5. COMPATIBILITY FEATURES

The model 8 offers many software and hardware features that are compatible with Series 90 and enable you to proceed directly to your new system and continue normal processing operations.

1.5.1. Software Compatibility

Table 1–1 summarizes the areas of software compatibility between Series 90 and the model 8. Column 1 lists software features – language compilers, file access methods, ICAM programs, IMS configurations, and so forth. Columns 2–4 list the three migration paths you can follow when you move to the model 8. To check the degree of compatibility for a particular feature, locate the entry for that feature under the migration path you intend to use. Where applicable, we've indicated the action you must perform to migrate that feature to the model 8.

Table 1–1. Series 90 – Model 8 Compatibility Features (Part 1 of 2)

Software Feature	Series 90 BDM to model 8 BDM	Series 90 CDM to model 8 CDM	Series 90 BDM to model 8 CDM
Assembler	Load code compatible	Load code compatible	Requires translation to CDM
BASIC	Source code compatible	Source code compatible	Source code compatible
ANSI 1968 COBOL	Load code compatible	N/A	N/A
ANSI 1974 COBOL	Load code compatible	Load code compatible	Source code compatible
ESCORT Programming Language	N/A	Load code compatible	N/A
FORTRAN IV	Load code compatible	Load code compatible	Source code compatible
RPG II	Load code compatible	Load code compatible	Source code compatible
DAM Files	Compatible	N/A	Requires translation to MIRAM

Table 1–1. Series 90 – Model 8 Compatibility Features (Part 2 of 2)

Software Feature	Series 90 BDM to model 8 BDM	Series 90 CDM to model 8 CDM	Series 90 BDM to model 8 CDM
SAM Files	Compatible	N/A	Requires translation to MIRAM
IRAM Files	Compatible	N/A	Compatible
ISAM Files	Compatible	N/A	Requires translation to MIRAM
MIRAM Files	Compatible	Compatible	Compatible
SAT Libraries	Compatible	Compatible	Compatible
MIRAM Libraries	N/A	Compatible	N/A
ICAM CUPS	Source code compatible	Source code compatible	Source code compatible
ICAM Generation	Source changes required	Source changes required	Source changes required
IMS Action Programs	Load code compatible	Load code compatible	Requires recompiling
IMS Single-thread	Source code compatible	Source code compatible	Source changes required
IMS Multithread	Source code compatible	Source code compatible	Source changes required
IMS Data Definitions	Load code compatible	Load code compatible	Source code compatible
Job Control Language*	Compatible	Compatible	Compatible
SORT Parameters	Load code compatible	Load code compatible	Source code compatible
Screen Formats	N/A	Release 7.1 formats require conversion	N/A

* The only changes you must make to your job control streams when you migrate to the model 8 are to specify new devices such as the 8470 disk or streaming tape.

1.5.2. Device Compatibility

Many Series 90 peripheral devices are also supported on the model 8. You can take many of your disks and tape devices as well as a variety of printers and card devices with you when you migrate. Here's a complete list of the devices that are compatible between systems:

- Disks

  8416, 8418, 8430, and 8433

  NOTE:

  8416 and 8418L disks cannot be used as system resident (SYSRES) volumes on the model 8.

- Tapes

  UNISERVO 10, 12, 14, 16, and 20

- Printers

  0770 and 0776

- Card Reader

  0716

1.6. MODEL 8 SUPPORT FOR 8470 DISK UNITS

The model 8 8470 disk supports BDM and CDM. Therefore, you don't have to bring your Series 90 disks forward to the model 8 to run applications under BDM. You can simply copy your Series 90 software to tape or diskette, load it on a model 8 8470 unit, and continue normal processing operations.

1.6.1. Moving Files to the 8470 Disk

The 8470 disk has much greater capacity than any Series 90 disk. Therefore, you may decide to move the contents of several Series 90 disks onto one 8470 disk. To increase system throughput and decrease response time, you should place heavily accessed files next to one another on the 8470 disk. To do this, you must first know your application environment and be able to identify those files you use most frequently.

Series 90 users of release 7.0.1 forward can use the system activity monitor (SAM) to provide detailed disk and file information. If you are migrating from releases 7.0.1 or 7.1, you can use SAM to identify heavily accessed Series 90 disks and those that have large average arm movements. For release 8.0 forward, use SAM trace mode and the volume table of contents (VTOC) of your Series 90 disks to identify heavily accessed files on those disks. This data should help you determine the file placement to optimize load balancing and minimize disk arm movement. For more information about SAM, see the OS/3 system activity monitor user guide/programmer reference, UP-8812 (current version).

Once you've migrated to the model 8, you can use the file placement analyzer (FIPLAN) to determine optimum file placement. FIPLAN is a general purpose performance tool that uses SAM trace data as input and projects an optimized disk file allocation. You can then allocate files to provide balanced loading across disks and ensure reduced seek time (time spent searching for files) within each disk. For more information about FIPLAN, see the file placement analyzer user guide/programmer reference, UP-9731 (current version).

1.6.2. Sector Size Effect on Performance

The physical sector size of the 8470 disk is 1024 bytes, but data may be accessed in either 1024- or 256-byte increments; each 1024-byte sector can be treated as four logical 256-byte sectors. When you access the 8470 disk in 256-byte logical sector increments, additional I/O requests are incurred when:

- the starting logical sector number does not fall on a 1024-byte physical sector boundary; or
- the number of bytes accessed is not a multiple of 1024.

Because the 8470 disk provides access to files in 256-byte mode, you can create files and subsequently access them with your existing Series 90 application programs without modifying these applications. However, when you develop new applications or update existing ones, you may want to modify them to use the 1024-byte sector size by increasing the I/O buffer size specification to a multiple of 1024. The buffer size requirements differ for each file type (SAM, DAM, ISAM, or MIRAM) and record type (fixed or variable format) being used by an application program. You can calculate the minimum buffer size required for a given record size by substituting either 256 or 1024 for the sector size variable. Refer to the OS/3 basic data management user guide, UP-8068 (current version), for details on buffer sizing calculations for SAM, DAM, and ISAM files. See the OS/3 consolidated data management concepts and facilities, UP-8825 (current version), for details on buffer sizing calculations for MIRAM files.

When you define file characteristics with a higher level language, the buffer size can be defined as a multiple of 1024:

- In COBOL, use the CONTAINS clause to specify either the number of records in a logical block or the number of bytes in a logical block.
- In RPG, use the BLOCKSIZE specification to define the number of bytes contained in a logical block.
- In FORTRAN, use the unit definition to define a block size.

By using a 1024-byte value for sector size in these calculations, an effective buffer size module of 1024 can be specified.

In general, accessing files in 1024-byte sector increments provides for the most efficient I/O operation, but may result in unused disk space. For example, an ISAM file with a 1200-byte block size requires two 1024-byte sectors per block, or 2048 bytes. If you access files in 256-byte sector increments, you can often save space but you may incur additional I/O requests. The same 1200-byte block size ISAM file requires five 256-byte blocks, or 1280 bytes. Although you save 768 bytes (you use 1280 bytes instead of 2048) when you use 256-byte sector increments, you also require five I/O requests instead of two.

Here are the accessing modes for the various file structures supported by the 8470 disk:

- File Formats

  You can maintain both system files and data files on the 8470 disk. System access technique (SAT), SAM, DAM, ISAM, and MIRAM files are all supported.

- System Files

  System files are maintained as SAT files and are based on a 256-byte sector format. The system access technique, and products that use SAT for logical I/O, use the 256-byte sector mode when accessing the 8470 disk. However, when multiple 256-byte blocks are accessed in one I/O, and the I/O begins on a 1024-byte sector boundary, no performance loss occurs.

- Data Files

  SAM, DAM, and ISAM files use the 256-byte sector mode. These files define a block-size value that represents the physical size of a block containing one or more records. This block size cannot be modified once the file is created. However, when the specified block size is rounded up to the next 256-byte boundary and this figure is a 1024-byte multiple, I/O is performed in 1024-byte increments.

  MIRAM files are accessed in either 1024-byte or 256-byte sector mode, depending on the size of the I/O buffer available in the application. If the application provides a buffer that is a multiple of 1024, and the buffer size is adequate to process records in 1024-byte sector increments, MIRAM performs read/write operations in 1024-byte increments. Otherwise, MIRAM performs read/write operations in 256-byte sector increments. In either mode, MIRAM records span sectors as necessary; no space is left unused between logical record slots.

1.7. MIGRATION AIDS

We provide several software aids to further simplify your migration effort. We'll tell you how to use these aids later; for now, here's a brief description of each:

- DTFCDI301

  The OS/3 DTF macro to OS/3 CDI macro converter (DTFCDI301) converts ASSEMBLER source code from DTF macrocode to CDI macrocode. DTFCDI301 converts file descriptions and I/O macros so they can run in a CDM environment.

- COBTRN303

  CDM doesn't support ANSI 1968 COBOL. Therefore, you must convert your ANSI 1968 COBOL programs to ANSI 1974 COBOL. We supply a program, COBTRN303, that converts ANSI 1968 COBOL source statements into a format acceptable as input to the ANSI 1974 COBOL compiler.

- SFCNVR

  The screen format conversion utility (SFCNVR) converts Series 90 screen formats so they can be used on the model 8. SFCNVR lets you modify complete screen format libraries or individual screen formats either interactively or in batch mode

1.8. AN APPROACH TO MIGRATION

The best way to begin your migration is to plan it. We suggest you review the flowchart shown in 1.2 and then develop an outline of the steps you'll follow during your own migration. Here are some other guidelines to help you carry out a successful migration:

1. Back up your system

   Be certain you have a current, backup copy of your Series 90 SYSRES volume before you attempt to migrate to the model 8. This is a standard precaution for anyone moving to a new computer system.

2. Use sample data files

   Use sample data files for program testing before you migrate all your data files to the model 8. This lets you identify any potential problems that could destroy complete data files.

3. Run applications side by side

   After you migrate to the model 8, continue running jobs on your Series 90 system while you test applications on the model 8. Parallel operation lets you thoroughly test your programs on the model 8 while using your Series 90 system as a backup. This prevents testing procedures or problems from completely halting your DP operations. We can't overemphasize the importance of running parallel production jobs.

4. Don't make changes during migration

   If possible, don't make any changes or enhancements to your programs while migration is under way. If you must make changes, monitor them carefully.

5. Have everything ready

   Have everything you need for the migration planned for before you begin. This includes personnel, documentation, system time, and recording medium.

6. Update your documentation

   Remember to update your in-house documentation (operator instructions, run books, standing orders, and so forth) to reflect your new data processing system.

2. Data File Migration

2.1. GENERAL

As we explained in Section 1, the model 8 is designed to operate primarily under CDM. CDM supports many new, interactive features such as screen format services, dialog processor, EDT, and enhanced workstation support. To use these features, you must convert the disk data files they use to MIRAM format.

In this section, we'll provide guidelines so you can decide which files you must convert, and show you how to convert those files to MIRAM.

2.2. MEDIA CONSIDERATIONS

Your data files can reside on a variety of media – disks, tapes, diskettes, and cards. However, only your disk data files must be converted to MIRAM. All tape, diskette, and card data files are compatible to both BDM and CDM, so you don't have to convert these files. Table 2–1 lists each type of media and shows which support BDM and which support CDM. We've broken down the disk media category into subcategories representing each type of disk file. As you can see, CDM supports MIRAM, IRAM, and SAT characteristic files only. All other disk data files must be converted to MIRAM to use CDM.

Table 2–1. Media Support for BDM and CDM

Media	BDM Support	CDM Support
Tape	Yes	Yes
Diskette	Yes	Yes
Card	Yes	Yes
Disk		
MIRAM	Yes	Yes
IRAM	Yes	Yes
SAT	Yes	Yes
SAM	Yes	No
DAM	Yes	No
ISAM	Yes	No
NI	Yes	No
ASAM	Yes	No

2.3. CONVERTING DISK DATA FILES WITH OS/3 DATA UTILITIES

We provide a program, the OS/3 data utilities, that lets you move your disk data files to the model 8 and convert them to MIRAM. All data utility job control streams are upward compatible from Series 90 to model 8 release 8.2. Thus, you can use data utilities to transfer your data files between peripheral devices (disk, tape, and diskette) either on Series 90 or on the model 8. You can also use data utilities to reformat your data files to your own specifications on either system.

The data utility program can be run in either batch or interactive mode. Interactive data utilities defaults to MIRAM for all output disk files; also, you don't need to include job control statements when you run interactive data utilities since all devices are assigned in the dialog. However, you must operate under CDM to use interactive data utilities because this routine does not execute in a BDM environment. You may not be familiar with interactive data utilities (which is supported only for release levels 7.1 forward), so we'll describe it briefly.

To use interactive data utilities, you key in RV I@DATA at your workstation. A series of questions in dialog format is then displayed on your workstation screen. Your answers to these questions specify which files you want to copy or reformat. If you don't understand a question, you can display a help screen that further defines the question. After you have answered all the questions, the DATA routine begins processing.

Regardless of whether you use data utilities in interactive or batch mode, this routine lets you copy your disk data files to tape or diskette on your Series 90 system. Then, you can load your data files to a model 8 8470 disk and reformat the data files, again using data utilities. Figure 2–1 illustrates this process.

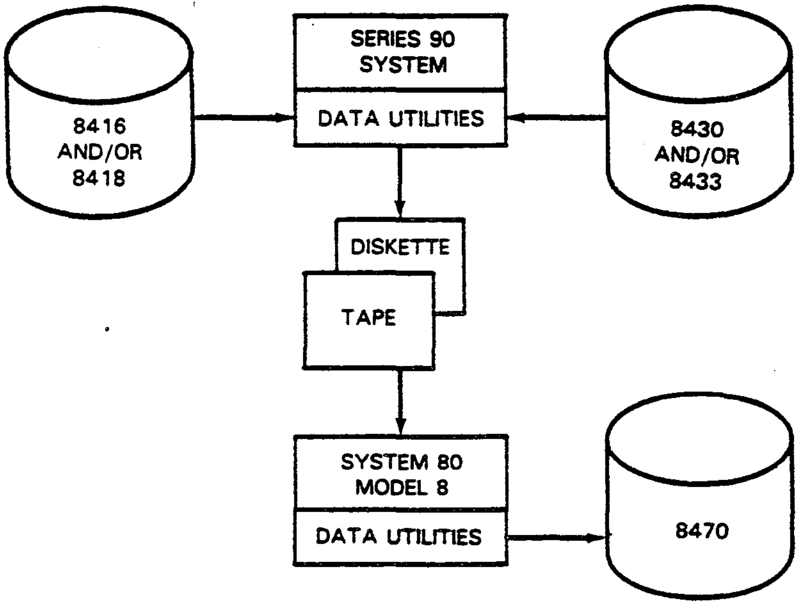


Figure 2–1. Copying Data Files Using Data Utilities

Since you can use batch data utilities in either data management environment, we've provided a sample execution just to give you an idea of how to run the program and which job control statements you are required to supply. You could use this job stream to restore a Series 90 data file contained on a diskette to a model 8 disk device.

2.4. SAMPLE EXECUTION OF BATCH DATA UTILITIES

The following job control stream shows a batch execution of data utilities in which a data file contained on diskette is transcribed to a MIRAM disk file. We execute the DATA routine in step 12. Steps 14A and 14B contain the program parameters that we use to format the output disk file to our specifications. Descriptions of each job control statement immediately follow the control stream; for more detailed explanations, refer to the OS/3 job control user guide, UP-9986 (current version).

```
1.   // JOB INDEX,,8000
2.   // DVC 20 // LFD PRNTR
3.   // DVC 130
4.   // VOL D01000
5.   // LBL FILEX
6.   // LFD INPUT1
7.   // DVC 50
8.   // VOL D01234
9A.  // EXT MI,C,0,CYL,4      (Use with 14A.)
or
9B.  // EXT MI,C,2,CYL,4      (Use with 14B.)
10.  // LBL XOUT
11.  // LFD OUTPUT1,,INIT
12.  // EXEC DATA
13.  /$
14A. UDD OM=(I,1,V),MK1=(10,0,DUP,CHG)
or
14B. UDD OM=(C)
15.  /*
16.  /&
17.  // FIN
```

Explanation:

1. The JOB statement assigns the name INDEX to the job and specifies the amount of main storage requested to run the job (8000 bytes in hexadecimal notation).

2. The DVC statement requests the assignment of a printer for the job. The logical unit number 20 means any available printer; the LFD name PRNTR is the system printer name and must be specified.

3–6. Input diskette device assignment statements, where:

3. The DVC statement requests the assignment of the diskette for the job.

4. The VOL statement indicates the diskette volume serial number (VSN).

5. The LBL statement shows the file name of the diskette file.

6. The LFD statement must have the logical file name INPUT1. This refers to the file identified in the preceding LBL statement (FILEX) and logically connects it to the data utility program executed in statement 12.

7–11. Device assignment for output disk file, where:

7. The DVC statement requests the assignment of your output disk pack.

8. The VOL statement indicates the volume serial number of the disk pack.

9A. (Use with 14A.) The EXT statement allocates disk space (extents) for an indexed disk file, where:

`MI`
The output file is a MIRAM disk file.

`C`
Disk space is allocated contiguously.

`Ø`
The file won't be extended if more disk space is needed.

`CYL`
Disk space is allocated by cylinders.

`4`
Four cylinders are requested initially.

9B. (Use with 14B.) This statement allocates disk space (extents) for a nonindexed disk file. The only difference between this statement and 9A is the 2, which indicates the file may be extended twice; only nonindexed files may be extended.

10. The LBL statement designates the output disk file name.

11. The LFD statement must have the logical file name OUTPUT1; the file must be initialized.

12. This statement executes the DATA routine.

13. This statement indicates the start of data utility parameters used to produce desired disk output file characteristics.

14A. Data parameters for indexed MIRAM output, where:

UDD
Parameter identifier for diskette-to-disk operation

OM
The output disk file is a MIRAM file.

I
The output disk file is indexed.

1
One 256-byte sector is assigned for the index buffer.

V
The output file is a multivolume mount file (required for indexed processing).

MK1
MIRAM output key 1 has the following characteristics:

1Ø
10-byte key

Ø
The key starts in relative record location 0.

DUP
Duplicate keys are allowed.

CHG
This key may be changed later.

14B. Data parameters for nonindexed MIRAM output, where:

UDD
Parameter identifier for diskette-to-disk operation

OM
The output disk file is a MIRAM file.

C
Nonindexed (sequential) output

15. This statement indicates the end of data utility parameters.

16. This statement signals the system that this is the end of the job.

17. This statement turns off the card reader.

2.4.1. MILOAD Utility

You may want to use a new utility program, MILOAD, when transferring your data files to the model 8. MILOAD can significantly increase performance when you reformat large, multikeyed MIRAM files. Your input files to MILOAD must reside on tape and must be created by OS/3 data management in EBCDIC mode. The output file created by MILOAD is always a MIRAM characteristic file. See the OS/3 data utilities user guide/programmer reference, UP-8834 (current version) for more information on how to use MILOAD.

2.5. IRAM/MIRAM CHARACTERISTICS

The MIRAM processor can access MIRAM files, IRAM files that it has created, or IRAM files created by the IRAM processor in BDM mode. Although CDM supports IRAM files, MIRAM offers certain advantages, and you may want to convert your IRAM files to MIRAM. To do so, use the OS/3 data utilities as described in 2.3. Here's a comparison of MIRAM and IRAM characteristics:

MIRAM	IRAM
■ More than one key per record	■ Only one key per record
■ Variable record lengths	■ Fixed record lengths
■ Can logically delete records (has record control byte)	■ Cannot logically delete records (no record control byte)
■ Duplicate record keys	■ No duplicate record keys
■ Allows key changes	■ No key changes
■ Minimum key length is one byte.	■ Minimum key length is three bytes.

Table 2–2 summarizes MIRAM/IRAM support for OS/3 components using BDM and CDM. Certain OS/3 components in release 6.1.2 and release 7.0/7.1 support limited MIRAM capabilities. Where IRAM ONLY is specified, only IRAM characteristic files are supported in that release.

Table 2-2. MIRAM/IRAM Component Support

Component	Release 6.1.2	Release 7.0/7.1	Release 8.0	Release 8.2
Assembler	BDM – MIRAM CDM – N/A*	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM
BEM BASIC	MIRAM not supported	MIRAM not supported	MIRAM not supported	Not available on model 8
Interactive BASIC	Not available	BDM – N/A CDM – Nonindexed MIRAM only	BDM – N/A CDM – Nonindexed MIRAM only	BDM – N/A CDM – Nonindexed MIRAM only
ANSI 1968 COBOL	MIRAM not supported	MIRAM not supported	MIRAM not supported	MIRAM not supported
ANSI 1974 COBOL	BDM – MIRAM CDM – N/A	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM
Data Utility	BDM – MIRAM CDM – N/A	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM
BEM Editor	BDM – SAT files only CDM – N/A	BDM – SAT files only CDM – N/A	BDM – SAT files only CDM – N/A	Not supported on model 8
Interactive Editor	Not available	BDM – N/A CDM – MIRAM data MIRAM library SAT library	BDM – N/A CDM – MIRAM data MIRAM library SAT library	BDM – N/A CDM – MIRAM data MIRAM library SAT library
ESCORT Programming Language	Not available	BDM – N/A CDM – MIRAM	BDM – N/A CDM – MIRAM	BDM – N/A CDM – MIRAM
FORTRAN IV	BDM – MIRAM CDM – N/A	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM
IMS	BDM – IRAM only CDM – N/A	Single thread BDM – IRAM only** CDM – MIRAM (single key) (no dup key) Multithread BDM – IRAM only** CDM – Not supported	Single thread BDM – MIRAM CDM – MIRAM Multithread BDM – MIRAM CDM – MIRAM	Single thread BDM – MIRAM CDM – MIRAM Multithread BDM – MIRAM CDM – MIRAM
RPG II	BDM – IRAM only CDM.– N/A	BDM – IRAM only CDM – MIRAM (single key)	BDM – IRAM only CDM – MIRAM	BDM – IRAM only CDM – MIRAM
SORT/SORT3	BDM – IRAM only CDM – N/A	BDM – IRAM only CDM – MIRAM	BDM – MIRAM CDM – MIRAM	BDM – MIRAM CDM – MIRAM

* N/A – Not applicable

** The IMS product accesses IRAM or MIRAM files with IRAM CHARACTERISTICS through MIRAM BDM.

2.6. DISK FILE SHARING

Often, several jobs try to access the same file at the same time. The OS/3 file lock feature lets you specify how a particular file is to be shared. If a file is not lockable, then it can be used by any job at any time.

To lock a file, specify the FILELOCK parameter in the SUPGEN phase during system generation. The FILELOCK parameter has two values:

```
FILELOCK={SHARE}
         {YES  }
```

where:

```
FILELOCK=SHARE
```

Is the default value and specifies that all files are lockable. This includes all system files (those prefixed with Y) and all user files.

```
FILELOCK=YES
```

Specifies that your supervisor maintains file lock capabilities for all system files and any user files prefixed with $LOK.

If you don't want certain files to be lockable, you must specify the YES value. Remember, however, that a nonlockable file is sharable without restriction.

For lockable files, you can specify how the file is to be shared (its share requirements) in the device assignment for the file. The share requirements determine how a program can use the file (read/write or read only access) and how other programs can concurrently use the file. A lockable file is sharable according to the following share requirements:

1. For sharability for a CDM accessed file, choose one of the following:

   a. Specify one of the following in the device assignment set of the program:

```
// DD ACCESS={EXC }
            {EXCR}
            {SRD }
            {SRDO}
            {SADD}
            {UCP }
```

where:

```
ACCESS=EXC
```

The program that declares this specification has exclusive read/write use of the file. The file cannot be shared.

ACCESS=EXCR

The program that declares this specification has read/write use of the file and allows other programs to share the file for read purposes only. This specification defines a single writer/multiple reader share environment. The only compatible specification is ACCESS=SRD.

ACCESS=SRD

The program that declares this specification is only permitted to read data from the file. Other programs can share the file for read/write purposes. Compatible options are ACCESS=EXCR, SRDO, and SADD.

ACCESS=SRDO

The program that declares this specification is only permitted to read data from the file. It lets other programs share the file for read purposes only provided they specify ACCESS=SRD or ACCESS=SRDO.

ACCESS=SADD

(For a CDM-only environment) The program that declares this specification has read/write use of the file and allows other programs to share the file regardless of whether they require read or read/write use. A compatible option is ACCESS=SRD.

ACCESS=UCP

The program that declares this specification allows the file to be shared for read/write purposes without restriction. No locks are set on the file; thus, several users can simultaneously access the same record for update, and updates may be lost as a result. If you specify the UCP option, you are responsible for maintaining file protection. This specification is not supported by IMS.

b. Specify // LFD *filename in the device assignment set of the program. (See 2.6.1 for information on the LFD job control statement).

c. Let the ACCESS keyword parameter default in the program's file definition.

2. For sharability for a BDM accessed file, choose one of the following:

a. Specify one of the following in the device assignment set of the program:

```
// DD ACCESS=⎧EXC ⎫
              ⎪EXCR⎪
              ⎨SRD ⎬
              ⎪SADD⎪
              ⎩UCP ⎭
```

NOTE:

For descriptions of these parameters, see 1a.

b. Specify // LFD *filename in the device assignment set of the program. (See 2.6.1 for information on the LFD job control statement.)

c. Specify the LOCK=NO parameter in the DTF macroinstructions for an ASSEMBLY language program.

d. Let ACCESS and LOCK keyword parameters default in the program's file definition.

NOTE:

In release 7.0 forward, the ACCESS defaults for both CDM and BDM are EXC for write operations and SRDO for read operations.

Table 2–3 summarizes how you can use each ACCESS parameter.

Table 2–3. Access Parameter Usage

// DD Access Parameter Specification	Use Permitted	
	Current Program	Other Programs
EXC	Read/Write	None
EXCR	Read/Write	Read
SRD	Read	Read/Write
SRDO	Read	Read
SADD (CDM only)	Read/Write	Read/Write
UCP (CDM only)	Read/Write	Read/Write

2.6.1. LFD Job Control Statement

You can place an asterisk (*) in front of the logical file name specified in the LFD job control statement to designate that this file is a read-only file. The // LFD *filename facility overrides the ACCESS parameter regardless of whether you specify this parameter in the program or in a // DD job control statement.

In release 6.1.2, the // LFD * statement is equivalent to SRD sharability. In release 7.0 forward, the // LFD * statement is equivalent to specifying SRDO. If you want to achieve release 6.1.2 share compatibility, remove the * and use the // DD ACCESS=SRD job control statement in the device assignment set for this file.

2.6.2. LOCK Keyword Parameter

The LOCK=NO keyword parameter is defined in a DTF file definition. This keyword is not valid for a CDM environment.

LOCK=NO

This is equivalent to specifying ACCESS=SRDO. Do not use it in the same DTF as the ACCESS keyword parameter. If you specify both parameters, the ACCESS parameter overrides the LOCK parameter.

If you omit both the LOCK=NO and ACCESS keyword parameters, this is equivalent to specifying ACCESS=EXC.

3. Source Program Migration

3.1. GENERAL

Your Series 90 programs are fully compatible with the model 8 at the load code level. Therefore, you can run all of your current programs on the model 8 without making any changes to them. We recommend, however, that when you migrate to the model 8, you convert any programs that use BDM to use CDM instead. As we explained in 1.3, you can't incorporate any new model 8 features into your existing programs until you convert those programs to use CDM.

In this section, we'll tell you how to convert your Series 90 programs that use BDM so they can run on the model 8 using CDM. We provide guidelines for converting programs written in each of the following languages:

- COBOL
- FORTRAN IV
- Report Program Generator II (RPG II)
- BASIC
- Basic Assembly Language (BAL)
- ESCORT Programming Language

In most cases, you can obtain CDM support simply by recompiling your programs. All procedures for compiling programs are fully compatible between systems. That is, you can recompile (or convert and reassemble for BAL programs) your programs on the model 8 just as you do on Series 90. And, since you can already run all of your current programs on the model 8 using mixed mode support, you don't have to recompile all your programs at once; instead, you can convert them at your convenience, according to your priorities and workload.

3.2. MOVING YOUR PROGRAMS TO THE MODEL 8

You can move your programs to the model 8 in two different ways:

1. Carry your Series 90 disks forward and use them on the model 8; for a complete list of all devices that are compatible between Series 90 and the model 8, see 1.5.2.

2. Copy your programs to a temporary media on your Series 90 system and restore them to a model 8 8470 disk device.

If you decide to transfer your programs to an 8470 disk, you must first copy your programs to tape or diskette on your Series 90 system; then, you can load them onto an 8470 disk on the model 8. We provide two librarian utilities – the SAT (system access technique) librarian and the MIRAM librarian – to help you accomplish this task. You use the SAT librarian to transfer programs contained in SAT library modules; use the MIRAM librarian for programs contained in MIRAM library modules.

The SAT and MIRAM librarians are fully compatible between systems. You can use the librarians to copy program elements on the model 8 just as you do on Series 90. You execute each of the librarian programs in the same way either on Series 90 or on the model 8. Also, you can use the same librarian parameters on either system.

In addition, when you transfer programs to an 8470 disk device, you don't have to worry about converting from BDM to CDM right away. The 8470 supports both BDM and CDM mode so you can copy your programs to an 8470 and continue normal operations on the model 8.

3.3. MIGRATING TO A CDM ENVIRONMENT

Here are some general guidelines for converting your programs to run in a CDM environment:

Language compilers generate data management interfaces according to the value you specify for the DMGTMODE parameter in the SUPGEN section of your supervisor. There are two possible values you can specify for DMGTMODE:

```
DMGTMODE={MIXED}
         {CDI  }
```

To run programs that haven't been converted to CDM, you must specify DMGTMODE=MIXED in your supervisor. (This is the default value.) Mixed mode supports both BDM and CDM; you can run all your programs in mixed mode on the model 8.

If you have converted your programs to CDM and want to run strictly in a CDM environment, specify DMGTMODE=CDI when you generate your supervisor. For more information on the DMGTMODE parameter, see the system installation user guide/programmer reference, UP-8839 (current version).

3.4. COBOL MIGRATION

3.4.1. Converting ANSI 1968 COBOL Using COBTRN303

Your ANSI 1968 COBOL programs can run in mixed mode on the model 8 without change. However, CDM doesn't support ANSI 1968 COBOL. Therefore, you must convert your ANSI 1968 COBOL programs to ANSI 1974 COBOL and then recompile those programs to run on the model 8 under CDM.

We provide a language converter, the COBOL converter (COBTRN303), that converts ANSI 1968 COBOL source code to ANSI 1974 COBOL source code. COBTRN303 also flags (issues a diagnostic message) any source statements that it cannot convert; you must convert these statements manually. COBTRN303 produces converted output that is acceptable to the ANSI 1974 COBOL compiler, except for those items that have been flagged. For more details on using COBTRN303, see the COBTRN303 user reference, UA-0314 (current version).

Here are some additional guidelines for converting ANSI 1968 COBOL to ANSI 1974 COBOL:

1. Data file support:

   You must convert all direct access method (DAM) data files to MIRAM format using OS/3 data utilities. We also recommend that you convert all SAM and ISAM data files to MIRAM, but this is necessary only if you specify DMGTMODE=CDI in your supervisor. The ORGANIZATION clause in your COBOL program is automatically changed if you use COBTRN303. If you convert your COBOL source code manually, you must make the changes shown in Table 3–1.

Table 3–1. Changes to ORGANIZATION Clause

File Type	ORGANIZATION Clause ANSI 1968 COBOL	ORGANIZATION Clause ANSI 1974 COBOL	DMGTMODE= MIXED	DMGTMODE= CDI
SAM	Sequential	SAM	Supported	Not supported
DAM	Relative	N/A	Supported	Not supported
ISAM	Indexed	ISAM	Supported	Not supported
MIRAM	N/A	Sequential, relative, or indexed	Supported	Supported

2. Features not supported by ANSI 1968 COBOL:

   - Consolidated data management
   - MIRAM data files
   - Workstations

3. Main storage requirements:

   ANSI 1968 COBOL is statically linked; ANSI 1974 COBOL can be statically or dynamically linked and may require increased main storage allocation on the // JOB statement.

4. Printer requirements:

   ANSI 1968 COBOL directs SYSLST to the printer. ANSI 1974 COBOL directs SYSLST to the system log file.

5. SEARCH ALL statement:

   In the SEARCH ALL statement, the indexed table item must be on the left side of the equal sign (=) in the WHEN clause.

6. Job control:

   The // LFD filename,,EXTEND statement is not supported for MIRAM files. Instead, specify OPEN EXTEND when you open a MIRAM file in your program.

3.4.2. Converting ANSI 1974 COBOL

Your BDM ANSI 1974 COBOL programs can run in mixed mode on the model 8 without any changes. To obtain CDM support, you must recompile your programs in CDM mode (specify DMGTMODE=CDI during supervisor generation). Here are some additional guidelines for converting ANSI 1974 COBOL:

1. ANSI 1974 COBOL data file support:

   You must convert all direct access method (DAM) data files to MIRAM format using OS/3 data utilities. We also recommend that you convert all SAM and ISAM data files to MIRAM, but this is necessary only if you specify DMGTMODE=CDI in your supervisor. Table 3–1 shows the changes you must make to the ORGANIZATION clause in your program.

2. Features not supported by ANSI 1974 COBOL:

   SAM and ISAM data files (if program is run in CDM mode)

3. Compiler main storage requirements (exclusive of job prologue):

Release	Main storage requirements
6.1.2	X'C800'
7.0, 7.1	X'D000'
8.0, 8.2	X'E000'

3.5. FORTRAN

3.5.1. BASIC FORTRAN

You can run your Series 90 BASIC FORTRAN programs on the model 8 in mixed mode. However, BASIC FORTRAN doesn't support CDM or MIRAM data files. To use CDM, you must convert your BASIC FORTRAN source code to FORTRAN IV and any data files to MIRAM.

3.5.2. FORTRAN IV

To convert to CDM mode, you must recompile your FORTRAN IV programs (specify DMGTMODE=CDI during supervisor generation) and make the following changes:

1. All disk data files must be converted to MIRAM format.

2. Unit definitions must be reassembled and relinked with updated definitions.

3. FORTRAN IV compiler and object modules must be placed in the proper order for CDM (module first in the library).

We deliver two sets of FORTRAN IV compiler macros in the system macro library ($Y$MAC), and two sets of FORTRAN IV compiler object modules in the system object library ($Y$OBJ). One set, called MIX, supports both BDM and CDM. The other set, called DTF, supports only BDM. The DTF modules are placed first in their respective libraries so the normal FORTRAN IV default is to BDM.

To support CDM, you must reverse the order of FORTRAN IV modules in $Y$MAC and $YSOBJ before assembling a unit definition or linking a FORTRAN IV object program. To do so, enter the following command to run the prefiled job stream MIXFOR4:

```
RV MIXFOR4
```

If you don't place these modules in the proper order when using the FORTRAN IV compiler run-time modules, errors will result.

If you want to use both BDM and CDM data management (mixed mode), and both FORTRAN compilers, copy the DTF procs to an alternate library before using the BASIC FORTRAN compiler. Then, supply the assembler with the name of the alternate library that contains the DTF procs whenever you assemble the unit definition for a BASIC FORTRAN program. Use the LIN parameter on the // PARAM statement during assembly. See the FORTRAN IV programmer reference, UP-8814 (current version) for information on using the LIN parameter.

NOTE:

In CDM mode, specify the FDIAGNOSE=YES parameter only for a printer, not for a workstation.

3.6. RPG II

Your BDM RPG II programs can run on the model 8 in mixed mode without modification or recompilation. To convert your RPG II programs to CDM mode, you must recompile your programs (specify DMGTMODE=CDI during supervisor generation), and change the disk data files your programs use to MIRAM format.

In a mixed mode environment, all disk file interfaces default to BDM. To generate CDM MIRAM interfaces, you must supply one of the following statements in the job control stream that executes the program:

```
// PARAM MIRAM=file1,file2,...,filen
```

or

```
// PARAM MIRAM=ALL
```

where:

`file1,...,filen`
Specifies which files are MIRAM; files not specified are accessed through BDM data management (specify DMGTMODE=MIXED during supervisor generation).

`ALL`
Specifies that all disk data files are MIRAM and are accessed in CDM mode (specify DMGTMODE=MIXED or DMGTMODE=CDI during supervisor generation).

NOTE:

These parameters are used only with disk data files. Other devices use CDM in both CDM and MIXED modes.

The // PARAM MOD=IRAM statement used in release 6.1.2 specifies that all IRAM files are accessed through BDM when you compile your programs in mixed mode. If you want your IRAM files accessed in CDM mode, replace the // PARAM MOD=IRAM statement with the // PARAM MIRAM=ALL statement.

You compile your RPG II programs on the model 8 just as you do on your Series 90 system. However, the RPG II compiler may require more main storage on the model 8; if necessary, increase the main storage allocation on the // JOB statement of the job that compiles the program. For more details on using RPG II on the model 8, see the RPG II user guide, UP-8067 (current version).

3.7. BASIC

BEM BASIC is not supported on the model 8. Instead, we provide a functionally equivalent product, interactive BASIC, which you can run in either CDM or mixed mode. You may already be familiar with interactive BASIC, since it is supported on Series 90 for releases 7.0.1 forward. To use interactive BASIC, log on to one of your model 8 workstations and enter the following command:

```
   BASIC
```

When the message BASIC READY appears on your workstation screen, you can begin entering the BASIC statements that comprise your program. Each line is checked for correct syntax as it is entered.

BEM BASIC and interactive BASIC are compatible with one exception: The workstation access method (WSAM), which is used by interactive BASIC, adds its own control character to a record for proper positioning. Therefore, when you create screens, you should output your screen with one PRINT command. If you build your screen with more than one PRINT command, WSAM will position the cursor in between the PRINT commands, which may affect your output.

Here's a summary of interactive BASIC features:

- Workstation support
- Consolidated data management (CDM)
- MIRAM data file support
- Extended HELP message processing

For more information on interactive BASIC, see the OS/3 BASIC programmer reference, UP-9168 (current version).

3.8. BASIC ASSEMBLY LANGUAGE (BAL)

You can run your BAL programs in mixed mode on the model 8 without making any changes. However, to convert to CDM mode, you must modify your source code and reassemble and relink your programs. You must also change any data files these programs use to MIRAM format. If your programs process SAM or DAM user disk labels, you must update them because MIRAM doesn't support user disk labels.

We provide a BAL macro converter, the OS/3 DTF macro to OS/3 CDI macro converter (DTFCDI301), that converts file definition and I/O macros from DTF macro code to CDI macro code. DTFCDI301 flags statements that can't be converted or that require special attention. It converts the following DTF macros:

DTFCD
DTFDA
DTFIR
DTFIS
DTFMI
DTFMT
DTFNI
DTFPR
DTFSD

In addition, the following imperative and other macros associated with these DTF macrcs are also converted:

CDIO
CLOSE
CNTRL
DPCA
DTFDM
ENDFL
ESETL
FEOV
GET
LBRET
MTIO
NOTE
OPEN
POINT
POINTS
PRIO
PRTOV
PUT
READ
RELSE
SAMIO
SETF
SETFL
SETL
SETP
SETS
TRUNC
WRITE

There are sufficient differences between DTF and CDI assembler specifications to preclude a 1-for-1 conversion approach. Therefore, we've provided the following guidelines to help your conversion effort.

3.8.1. BAL Conversion Guidelines

1. A program that successfully executes in a Series 90 release 7.0 forward environment can execute on the model 8 without reassembly. However, user physical I/O programs that execute with the system access bit in the CCB turned off aren't supported. This bit is reset by the CCB macro when the fourth positional parameter (error option) excludes an X'08' setting. You must modify programs that use this feature to a data management or system access level. Consider converting these programs to a higher level language.

2. Programs that operate at the physical I/O level must not be dependent on the particular sequence of operations used by a particular channel.

3. Programs should not use negative relocation for addressing unauthorized storage areas.

4. Programs must be independent of the relations between instruction execution times, I/O data rates, access times, I/O command execution times, and timing facilities including elapsed time values.

5. Programs must not use Series 90 low order storage locations assigned specifically for system use.

6. Programs must not depend on Series 90 machine dependent functions.

7. The following Series 90 privileged instructions don't exist on the model 8:

    a. SOFTSCOPE forward scan (SSFS)

    b. SOFTSCOPE reverse scan (SSRS)

    c. Load control storage (LCS)

8. Change any specific references to fields contained in DTF structures in the user region of your programs.

9. Contingency routine addresses for EOFADDR and ERROR aren't supported by CDM; access these addresses with inline code following imperative commands.

3.8.2. DTF ISAM to CDI MIRAM Keyword Conversion

Table 3–2 describes the differences between DTF keyword parameters and CDI keyword parameters. Look at this table first, and then at the conversion that follows it. In the conversion, a DTF ISAM file is converted to a CDI MIRAM file with a single key. Note that line sequence numbers are used for comparison between the DTF and CDI parameters.

Table 3–2. Description of DTF and CDI Keyword Parameters (Part 1 of 2)

Sequence Number	DTF Parameter	DTF Description	CDI Parameter	CDI Description
1	ACCESS	Specifies how the file is to be shared	ACCESS	Specifies how the file is to be shared
2	BLKSIZE	Block size calculated by (RECORD LENGTH + 5) x (NUMBER OF RECORDS PER BLOCK)	BFSZ	Buffer size calculated on record and sector sizes as defined for MIRAM files
3	ERROR	ERROR contingency routine address	-	No corresponding CDI parameter
4	INDAREA	Main storage location for index blocks during load operations, or for top index table during random retrieval or record insertion	INDA	Same function as in DTF mode

Table 3–2. Description of DTF and CDI Keyword Parameters (Part 2 of 2)

Sequence Number	DTF Parameter	DTF Description	CDI Parameter	CDI Description
5	INDSIZE	Length of index area	INDS	Length of index area
6	IOAREA1	Location of I/O area	IOA1	Location of I/O area
7	IOROUT	Type of file processing	–	No corresponding CDI parameter
8	KEYARG	Program location for addresses/keys to affect record retrieval	KARG	Same as in DTF mode
9–10	KEYLEN KEYLOC	Length and offset of record key	KEYn= (LEN,LOC)	Length and offset of record key
11	PCYLOFL	Percentage of cylinder overflow	–	No corresponding CDI parameter
12	RECFORM	Record format	RCFM	Record format
13	RECSIZE	Maximum record length	RCSZ	Maximum record length
14	TYPEFLE	Sequential/random access	MODE	Sequential/random access
15	VERIFY	Specifies if output record parity is to be checked	VRFY	Specifies if output record parity is to be checked
16	WORK1	Location of the record work area	WORK	Specifies that records will be processed in a work area instead of in a buffer
17	–	No corresponding DTF parameter	PROC	Keyed/unkeyed processing

Here are the characteristics of the DTF ISAM file before conversion to MIRAM:

Parameter		Value	Sequence Number
DTFIS ACCESS	=	EXC	1
BLKSIZE	=	427	2
ERROR	=	errtn	3
INDAREA	=	indx	4
INDSIZE	=	256	5
IOAREA1	=	newrec	6
IOROUT	=	LOAD	7
KEYARG	=	ssnno	8
KEYLEN	=	9	9
KEYLOC	=	39	10
PCYLOFL	=	2Ø	11
RECFORM	=	FIXBLK	12
RECSIZE	=	8Ø	13
TYPEFLE	=	RANSEQ	14
VERIFY	=	YES	15
WORK1	=	worka	16

To convert the DTF ISAM file to a CDI MIRAM file with a single key, we make the following changes:

	Parameter		Value	Sequence Number
RIB	ACCESS	=	EXC	1
	BFSZ	=	512	2
	INDA	=	indx	4
	INDS	=	256	5
	IOA1	=	newrec	6
	KARG	=	ssnno	8
	KEY1	=	(9,39,NDUP,NCHG)	9–10
	RCFM	=	FIX	12
	RCSZ	=	80	13
	MODE	=	RAN	14
	VRFY	=	YES	15
	WORK	=	YES	16
	PROC	=	KEY	17

3.8.3. DTF to CDI Processing Considerations

Here are some considerations when converting from DTF to CDI:

1 Opening files:

To open a file in DTF, use:

```
OPEN INP
```

To open a file in CDI, use:

```
OPEN INP,(INPRIB)
```

You must supply the resource information block (RIB) name when you open a file using CDI. This sets up the file structure; otherwise, all defaults for the file structure are applied.

2. Closing files:

To close a file in either DTF or CDI, use:

```
CLOSE INP
```

3. Sequential processing:

To sequentially process an ISAM file in DTF, use:

```
SETL INP,BOF
```
(This sets file at beginning.)

To continue processing, use:

```
GET INP,WORKA        (This reads a record sequentially.)
PUT INP,WORKA        (This updates the record.)
```

To terminate sequential processing, use:

```
ESETL INP
```

To sequentially process a MIRAM file, you must first be sure you are starting at the beginning of the file. If you have just issued the OPEN macro for that file, the current location is automatically set to the beginning of the file. If the file has not just been opened, use the following macro to set the current position to the beginning of the file:

```
DMSEL INP,RECORD,BOF           (The BOF option places you at the
                               beginning of the file.)
```

To continue processing, use:

```
DMINP INP,WORKA,,,SEQ          (This reads a record sequentially.)
DMUPD INP,WORKA                (This updates the record.)
```

NOTE:

We supply the SEQ parameter in the DMINP macro to override the MODE=RAN parameter in line 14 of the RIB. (See Table 3–2.)

4. Random processing:

To randomly process the file in DTF, specify the keyword parameter IOROUT=ADD or IOROUT=ADDRTR; then, use the following:

```
WRITE INP, NEWKEY
```

or (Use either macro to write a new record.)

```
ADD INP
WAITF              (This delays processing until prior action terminates.)
```

```
    READ INP,KEY
```

and (This retrieves record by location.)

```
    WAITF
    WRITE INP,KEY
```

or (This rewrites the last record retrieved.)

```
    UPDT INP
```

followed by:

```
    WAITF
```

To randomly process a file in CDI, use:

`DMOUT INP,WORKA`	(This writes a new record.)
`DMINP INP,WORKA`	(This retrieves a record by key.)
`DMUPD INP,WORKA`	(This updates a record.)
`DMDEL INP`	(This deletes a record.)

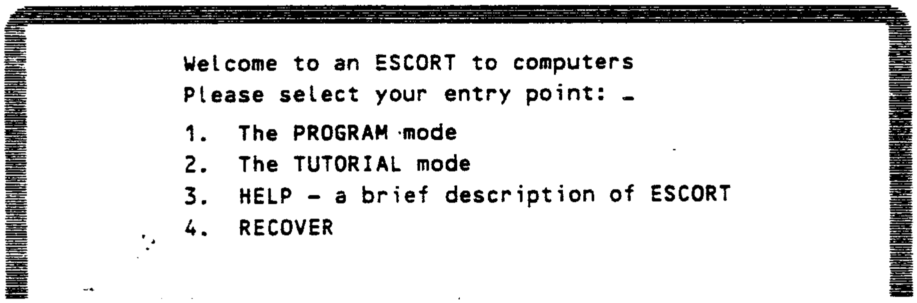
3.9. ESCORT PROGRAMMING LANGUAGE

The ESCORT programming language uses CDM interfaces and is completely compatible between Series 90 and the model 8. You don't have to make any changes to your ESCORT programs or data files to use them on the model 8. Since you may not have used the ESCORT programming language before, we'll tell you more about it now.

The ESCORT interactive programming language uses English statements to create a program. You can generate reports and perform inquiry and update routines through the use of simple, sentence-like programs, entered through your workstation. To use the ESCORT programming language, log on and then enter the following command:

```
ESCORT
```

The following display appears on your workstation screen:



```
Welcome to an ESCORT to computers
Please select your entry point: _
1.  The PROGRAM mode
2.  The TUTORIAL mode
3.  HELP - a brief description of ESCORT
4.  RECOVER
```

Figure 3–1. Initial ESCORT Screen

For an inexperienced user, the ESCORT programming language offers a tutorial mode of operation so you can learn the language right at your workstation. It also offers HELP screens to explain the various statements and conventions it requires. For more detailed information on the use of the ESCORT programming language, refer to the OS/3 ESCORT programming language user guide, UP-8855 (current version).

4. System Generation

4.1. GENERAL

There are physical differences between your Series 90 system and the model 8. Therefore, you must adjust the model 8 system generation parameters before you can use Series 90 software on your new system.

You can perform system generation (SYSGEN) on the model 8 just as you do on Series 90. The model 8 supports the SYSGEN dialog, which lets you generate supervisors interactively at a workstation. Or, if you prefer using cards, you can also generate your model 8 supervisors in batch mode. Regardless of the method you use, all SYSGEN procedures are fully compatible between systems.

In addition, we deliver a basic supervisor, SY#BAS, as part of the system control software. SY#BAS eliminates the need for you to perform a SYSGEN operation unless you have a specific reason to do so; for a complete list of SY#BAS features, refer to the model 8 system release description (SRD). If you require features not supplied by SY#BAS, you must generate a supervisor tailored to your own needs. The system installation user guide/programmer reference, UP-8839 (current version) describes the procedures for generating a supervisor either interactively or in batch mode.

4.2. SUPGEN PARAMETERS

The model 8 offers several new SUPGEN parameters that are not available with your Series 90 system. In addition, many parameters you've used on your Series 90 system have new default values on the model 8.

Table 4–1 lists SUPGEN parameters that are either new or have been changed since Series 90 release 6.1.2. The table provides the default values for each of these parameters for Series 90 releases 7.1 and 8.0 and for model 8 release 8.2. We've also provided a brief description of each parameter in the table; for more detailed descriptions, see the system installation user guide/programmer reference, UP-8839 (current version).

Table 4–1. New or Changed SUPGEN Parameters (Part 1 of 2)

Parameter	Description	Release Default		
		7.1	8.0	8.2
DDPSPOOL	Number of concurrent running tasks to return spoolout of DDP	N/A	10	10
DMGTMODE	Data management mode	DTF	DTF	Mixed
DYNBUFMGT	Dynamically allocates main storage	Mixed	None	Basic
ERRLOG	I/O error log	Yes	No	Yes
ERRLOGBUF	Number of error log buffers	3	3	6
EXPREGION	System expansion region size	4096	4096	4096
FILELOCK	Type of filelock system	No	No	Share
FLOATPT	Floating point support	No	No	Yes
IGNORESFT	Ignore // SFT statements	No	No	No
ISADMID	System security administrator ID	(No default – supply ID)		
ISBATCHLMT	Maximum number of batch sessions	8	8	8
ISINTLMT	Maximum number of interactive users	8	8	8
ISINTPRI	Interactive commands priority level	1	1	1
ISKEYS	Limits interactive services space in main storage	No	No	Yes
ISLOGONSC	Logon security	No	No	No
JOBACCTREQ	Job accounting	N/A	No	No
RECOVERDS	Disk space recovery option	No	No	No
RESBUFSIZE	Resident buffer pool size	500	500	500
RESHARE	Specifies resident shared load modules	(Supply module name)		
RESMOD	Software modules resident	NONE	NONE	SMSLOD
SAM	System activity monitor in supervisor	NO	NO	NO
SCDINDEX	Faster loading of shared code modules	N/A	NO	NO
SHAREDMGT	Number of 24-byte resident slots controlling shared data management	40	40	40
UNATCONSOLE	Unattended system console	N/A	N/A	0
VOLTABLE	System volume table made resident	N/A	NO	NO
COMM*	Number of communication lines (SLCAs) in the system	1	1	NO

Table 4-1. New or Changed SUPGEN Parameters (Part 2 of 2)

Parameter	Description	Release Default		
		7.1	8.0	8.2
CHAN*	Input/output microprocessor (IOMP) channel number	N/A	N/A	13
IORB*	Number of input/output resource blocks (IORBs) generated for channel	N/A	N/A	5 X COMM
COMM1*	Number of communication lines (SLCAs) in the system. Must specify if system supports two IOMP channels.	N/A	N/A	NO
CHAN1*	Second IOMP channel number	N/A	N/A	15
IORB1*	Number of IORBs generated for second channel	N/A	N/A	5 X COMM1

*SUPGEN communication (ICAM) parameters

4.3. SYSTEM FILE DESCRIPTION

Many new system files have been added to OS/3 release software since Series 90 release 6.1.2. Table 4-2 lists each new system file and describes the file type and the contents of each file.

Table 4-2. System File Description

File Name	File Type	Description
YMIC	SAT	System microcode library
YSCLOD	SAT	Shared code load library
YFMT	MIRAM	Contains the system screen formats and the menu generator screens and help screens
YSAVE	MIRAM	Saved run-library modules
YDIALOG	MIRAM	Contains system dialogs for SYSGEN and JCL
YSDF	MIRAM	Points to current microcode levels in the YMIC file that permits loading of microcode to the CPU and peripherals
YHELP	MIRAM	Contains help screen modules
YSEC	MIRAM	Contains security user profiles
YSJF	SAT	System journal file, used to record changes in hardware status
YESUM	MIRAM	System error log summary
IVPLIB	SAT	Installation verification program library
SMCFILE	SAT	Contains the system maintenance corrections (SMCs) processed by the system
YSMCLOG	MIRAM	Used for the system maintenance correction log (SMCLOG)

5. ICAM Migration

5.1. GENERAL

Series 90 integrated communications access method (ICAM) programs are source code compatible with the model 8. Therefore, you can run most of your communications user programs (CUPs) on the model 8 without change. But ICAM has been enhanced to include many new features since OS/3 release 7.0.1. These features include global network support, interactive services interface (DMI), and new workstation support. To use these new ICAM features in CUPs that currently run under BDM, you must convert those CUPs to use CDM and regenerate your ICAM network to define the model 8 communications (channel and line) network configuration.

In this section we'll tell you how to upgrade your existing CUPs to the model 8. It's really a very simple process. In most cases, all you have to do is recompile your CUPs on the model 8, just as you previously did on your Series 90 system. The only exception to this is communications physical interface (CPI) user programs, which require some additional changes. We describe these changes and also include sample network definitions so you can see how to incorporate new ICAM features into your user programs.

5.2. COMMUNICATIONS USER PROGRAMS (CUPS)

5.2.1. COBOL CUPS

Series 90 COBOL CUPs are source compatible with the model 8 and can run in a mixed mode data management environment without change. To use CDM interfaces for data files, you must recompile your COBOL CUPs on the model 8.

5.2.2. RPG II CUPS

Series 90 RPG II CUPs are source compatible with the model 8 and can run in a mixed mode environment without change. To use CDM interfaces for data files, you must recompile your RPG II CUPs on the model 8.

5.2.3. ASSEMBLER CUPS

Series 90 ASSEMBLER CUPs are source compatible with the model 8 and can run in a mixed mode environment without change. To use CDM interfaces for data files, you must convert and reassemble your ASSEMBLER programs on the model 8. See 3.8 for details on converting ASSEMBLER programs to CDM mode.

5.3. NETWORK DEFINITION

ICAM lets you:

- input data from a network of remote terminals into a program for processing;
- distribute messages to the terminals within the network; and
- transfer messages from terminal to terminal.

Generally, you have only one network defined and operating on your system. However, ICAM permits you to define several distinct networks and to have these networks operating simultaneously. You define each network during system generation (SYSGEN) by submitting a set of macroinstructions that define the terminals, lines, buffers, and queues for that network. In addition, you must specify which of the four available communications interfaces each network is to use. A terminal can be included in more than one network definition; however, only one network can use the terminal at a time.

Model 8 console messages and responses reference the network name instead of the jobslot. Also, the model 8 communications hardware configuration supports up to two channels with 14 lines (ports) on each channel. Your ICAM network generation must include the physical channel and line specification in the LINE and CACH macroinstructions.

Table 5–1 describes network definition parameters that are either new or have been changed for Series 90 OS/3 releases 6.1.2, 7.0.1/7.1 and for Model 8 release 8.2. Parameters shown with an asterisk (*) give an example of an acceptable keyword value. Those without an asterisk specify the default value for that parameter. See the ICAM network definitions and operations user guide, UP-8947 (current version) for more details on these parameters.

Table 5–1. Network Definition Parameters (Part 1 of 2)

Parameter	Release 6.1.2	Release 7.0.1/7.1	Release 8.2
CCA			
TYPE=(GBL,,node)		Global network	Global network
DCA=YES	N/A	Distributed communications	Distributed communications
DUSTERR=INLINE	N/A	N/A	DDI and STDMCP
FEATURES			
TRACEMAX	ICAM TRACE	ICAM TRACE	N. A. see ITF.
AUTOBUF	N/A	N/A	CPI only
MONITOR	SAM link	SAM link	SAM link
GAWAKE=YES	N/A	Global and dedicated	Global and dedicated

Table 5–1. Network Definition Parameters (Part 2 of 2)

Parameter	Release 6.1.2	Release 7.0.1/7.1	Release 8.2
BUFFERS			
EXPFACT=n - default is 25	N/A	N/A	Expansion factor
UDUCT=(20,25,3)	N/A	DCA and PDN	DCA and PDN
LINKPAK=(57,63,23)	N/A	DCA and PDN	DCA and PDN
RTIMER=50	N/A	DDP	DDP
LOCAP	Global network	Global network	Global network
TYPE=STDMCP	Interface type	Interface type	Interface type
*DUSTERR=INLINE	N/A	N/A	DUST error return
*LOW=filename (name associated with a DISCFIL macro)	Input queues	Input queues	Input queues
*MEDIUM=MAIN	Input queues	Input queues	Input queues
*HIGH=MAIN	Input queues	Input queues	Input queues
*REMOTE=(node)	N/A	N/A	PDN
JOBNAME=name of filed COBOL job stream		COBOL (CMCS)	COBOL (CMCS)
*JOBINIT=(LOAD,REPORT)		COBOL (CMCS)	COBOL (CMCS)
*IAS=(YES,OFF)	N/A	I/S DMI	I/S DMI
PROTYP=DICE	N/A	Edit type	Edit type
MODE=SYSTEM	N/A	Remote workstations	Remote workstations
MT=YES	N/A	Interactive services	N/A
LINE			
DEVICE=			
3271	IBM 3270 support	IBM 3270 support	IBM 3270 support
INV3271	N/A	N/A	3270 emulator
RWS	N/A	N/A	Remote workstations
CIRCWS=PDN-name	N/A	PDN	PDN
CHAN=13	N/A	N/A	Channel number
PGROUP (RWS)			
PGID=polling-group-rid	N/A	N/A	Remote workstations
PTIME=(1,60)	N/A	N/A	Remote workstations
TERM			
*(U400,1920,SBT,CP)	UTS 400	UTS 400, UTS 20, or UTS 40	UTS 400
*(U20,1920,SBT,CC,PRIMARY)	N/A	N/A	UTS 20 support
*(U40,1920,,PRIMARY)	N/A	N/A	UTS 40 support
*CALL=phone-number	N/A	DTE on PDN	DTE on PDN
*TCTUPD=YES	N/A	Interactive services	Interactive services
*REMOTE=NODE	N/A	N/A	PDN
*INONLY=YES	Send-only terminal	Send-only terminal	Send-only terminal
*OUTONLY=YES	Receive-only terminal	Receive-only terminal	Receive-only terminal
*DISIN=n	N/A	PDN	PDN
*DISOUT=n	N/A	PDN	PDN
INRECSZ=80	N/A	N/A	Batch terminal
*DVCGID=term-group-name	N/A	N/A	DCA with DCP
PROTYP=DEV	N/A	N/A	DCP
RBATCH=NO	N/A	N/A	DCP
SESSION			
*EU1=(name)	Static user	Static user	Static user
*EU2=(name)	Static user	Static user	Static user
VLINE		Two computer nodes	Two computer nodes
ID=ca-port-number		Global network	Global network
*DEVICE=(X25,SECONDARY)			
K=value			
RETRY=(n,m)			
TIMEOUT=(n,m)			
LBUF=(num,size,thresh)			
*TYPE=(9600)			
CMADDR=n			
RSPADDR=n			

5.4. ICAM NETWORKS

ICAM supports two types of networks: dedicated and global. A dedicated network can be accessed by only one program at a time. A global network permits several programs to access the network concurrently. Only one global network can be operating at any given time.

The following subsections contain two examples: first, a sample dedicated network definition, and then a sample global network definition. We'll tell you when to use each type of network and provide descriptions of the parameters defined in each network.

5.4.1. Sample Dedicated Network Definition

A dedicated network permanently assigns (dedicates) lines and terminals to one program at a time. Use a dedicated network when programs use a specific network of terminals that are defined locally, and never need to share a network of terminals at the same time.

Sample ICAM dedicated network definition:

```
① NET1     CCA       TYPE=(TCI),FEATURES=(OPCOM,OUTDELV)
                    BUFFERS   40,64,5,ARP=25
   LNE1     LINE      DEVICE=(UNISCOPE),TYPE=(9600,SYNC),ID=6,CHAN=13,
                      STATS=YES                                          X
   TRM1     TERM      FEATURES=(U400,1920),ADDR=(28,51),AUX1=(COP,73),   X
                      LOW=DQFILE1,MEDIUM=MAIN,HIGH=MAIN
   TRM2     TERM      FEATURES=(U400,1920),ADDR=(28,52),AUX1=(COP,73),   X
                      LOW=DQFILE1,MEDIUM=MAIN,HIGH=MAIN
   LNE2     LINE      DEVICE=(UNISCOPE),TYPE=(9600,SYNC),ID=8,CHAN=13,
                      STATS=YES                                          X
   TRM3     TERM      FEATURES=(U400,1920),ADDR=(29,51),                 X
                      LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
② PRF1     PRCS      LOW=DQFILE2
③ DQFILE1  DISCFILE  FILEDIV=8
④ DQFILE2  DISCFILE  FILEDIV=8
⑤ TCIDTF   DISCFILE  MSGSIZE=1920
            ENDCCA
            MCP       MCPNAME=C2
            CACH=(06,NET1,01)
            CACH=(08,NET1,02)
   END
```

where:

① Identifies NET1 as the TCI dedicated network name.

② Creates a process file to be used with disk file buffers for IMS.

③ Identifies file to be used for output disk file buffers.

④ Identifies file to be used with process file for unsolicited output.

⑤ Defines input buffer file, required for IMS.

5.4.2. Sample Global Network Definition

A global network allows multiple programs to concurrently use the same network and the same lines. You can use a single global network instead of multiple dedicated networks. A typical global network is one in which a user can sign on to IMS or to interactive services from the same terminal. Use a global network when:

- programs need to share the same terminals;
- the communications system has more than one central processor; or
- the communications system uses remote workstations.

Sample ICAM global network definition:

```
① NET1      CCA       TYPE=(GBL,,S),FEATURES=(OPCOM,OUTDELV),GAWAKE=YES   X
                      DCA=YES
            BUFFERS   100,64,5,ARP=100
② IMS1      LOCAP     TYPE=(TCI)
③ STDM      LOCAP     TYPE=(STDMCP)
④ INTS      LOCAP     TYPE=(DMI),IAS=(YES,OFF),MT=YES,MODE=SYSTEM
  LNE1      LINE      DEVICE=(UNISCOPE),TYPE=(9600,SYNC),ID=6,CHAN=13
  TRM1      TERM      FEATURES=(U400,1920),ADDR=(28,51),AUX1=(COP,73),    X
                      LOW=DQFILE1,MEDIUM=MAIN,HIGH=MAIN
  TRM2      TERM      FEATURES=(U400,1920),ADDR=(28,52),AUX1=(COP,73),    X
                      LOW=DQFILE1,MEDIUM=MAIN,HIGH=MAIN
  LNE2      LINE      DEVICE=(UNISCOPE),TYPE=(9600,SYNC),ID=8,CHAN=13
  TRM3      TERM      FEATURES=(U400,1920),ADDR=(29,51),                  X
                      LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
⑤ LNE3      LINE      DEVICE=(LWS),STATS=YES
  TRM5      TERM      FEATURES=(U20),ADDR=(C15),INPUT=PRF2,               X
                      LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
⑥ LNE4      LINE      DEVICE=(LWS),STATS=YES
  TRM6      TERM      FEATURES=(U20),ADDR=(C16),                          X
                      LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
⑦ LNE5      LINE      DEVICE=(RWS),TYPE=(4800,SWCH,SYNC),ID=10,CHAN=15,   X
                      RETRY=(6,4)
⑧           PGROUP    PGID=31
⑨ TRM7      TERM      FEATURES=(U40,,,PRIMARY),ADDR=(31,51),LOW=MAIN,     X
                      MEDIUM=MAIN,HIGH=MAIN,AUX1=(COP,73),AUX2=(TP,74)
  TRM8      TERM      FEATURES=(U40,,,SECONDARY),ADDR=(31,52),LOW=MAIN,   X
                      MEDIUM=MAIN,HIGH=MAIN,AUX1=(COP,73),AUX2=(TP,74)
⑪PRF1       PRCS      LOW=DQFILE2
⑫PRF2       PRCS      LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
⑬DQFILE1    DISCFILE  FILEDIV=8
⑭DQFILE2    DISCFILE  FILEDIV=8
```

```
          ENDCCA
          MCP       MCPNAME=C2
          CACH=(Ø6,NET1,Ø1),
          CACH=(Ø8,NET1,Ø2),
          CACH=(1Ø,NET1,Ø5),
     END
```

where:

① Specifies a single node global network using distributed communications architecture that includes operator communications and dynamic session capabilities.

② Specifies a locap file to support IMS. Locap-name (IMS1) must agree with locap name in the IMS configurator.

③ Specifies a locap file to support a standard interface user program (STDM).

④ Specifies a locap file to support interactive services. Interactive services uses the demand mode interface (DMI) to communicate with ICAM. In release 7.1, locap-name must agree with name specified in the SUPGEN parameter ISLOCAPID. In release 8.0, the ISLOCAPID parameter is not supported. MODE=SYSTEM must be specified to operate in system mode with remote workstations.

⑤ Defines a local workstation terminal. This terminal has a static session defined for CUP STDM and is not available to IMS.

⑥ Defines a local workstation that is available to all CUPs.

⑦ Specifies a remote workstation.

⑧ Specifies that the following workstations belong to this polling group (RID 31). This poll group uses a default value of 1 second for a normal poll interval and 60 seconds for a slow poll interval.

⑨ Specifies the remote workstation primary screen of a UTS 40 terminal. The default designates a screen size of 1920 characters.

⑩ Specifies the remote workstation secondary screen of a UTS 40 terminal.

⑪ Creates a process file to be used with disk file buffers.

⑫ Creates a process file.

⑬ Identifies file to be used for output disk file buffers.

⑭ Identifies file to be used with process file for unsolicited output.

NOTE:

Each line macroinstruction (except LWS) specifies the model 8 channel (CHAN=) and line (ID=) number.

5.5. REMOTE WORKSTATION SUPPORT

Your current Series 90 system may not offer remote workstation support, which is provided only for OS/3 release 8.0 forward. We'll describe this feature briefly here, and tell you how to define ICAM networks using remote workstations.

Remote UTS 20 and UTS 40 terminals can operate as if they were local workstations. However, UTS 20 and UTS 40 terminals connected to a cluster controller can't be used as remote workstations. Here's how you specify remote workstation parameters:

- In the SUPGEN phase of SYSGEN, specify:

`SCRENMEM=1`	For terminals without the screen bypass feature
`SCRENMEM=2`	For terminals with the screen bypass feature; always use this parameter for a UTS 40 terminal.

- When you define an ICAM network, specify the following parameters for each macroinstruction:

	Macro	Parameters
1.	`CCA`	`DCA=YES and GAWAKE=YES`
2.	`LOCAP`	`MODE=SYSTEM`
3.	`LINE`	`DEVICE=(RWS).` Note that all terminals defined on this line are remote workstations. `TYPE=(SYNC)`
4.	`PGROUP`	Poll group as defined by the RID of each terminal. All terminals on a poll group must have the same RID.

5.	`TERM`	UTS 20 remote workstations with single screen storage:
		`WS1 TERM FEATURES=(U20,1920),ADDR=(32,51),AUX1=(TP,73)`
		UTS20 remote workstation with dual screen storage:
		`WS2 TERM FEATURES=(U20,1920,,,PRIMARY),ADDR=(33,51), X`
		`AUX1=(TP,73),LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN`
		`WS3 TERM FEATURES=(U20,1920,,,SECONDARY),ADDR=(33,52),X`
		`AUX1=(TP,73),LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN`

NOTE:

The keywords PRIMARY and SECONDARY specify the primary and secondary display screens on a UTS 20 remote workstation. Remote workstations are defined in the TERM macroinstructions previously shown. The definitions are the same except that the first TERM macro must specify PRIMARY and the second macro must specify SECONDARY. In addition, the station address (SID) you specify for the secondary display screen must be 1 digit higher than that specified for the primary display.

UTS 40 remote workstation with dual screen storage:

```
WS4    TERM    FEATURES=(U40,1920,,PRIMARY),ADDR=(35,51),    X
               AUX1=(TP,73),LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
WS5    TERM    FEATURES=(U40,1920,,SECONDARY),ADDR=(35,52), X
               AUX1=(TP,73),LOW=MAIN,MEDIUM=MAIN,HIGH=MAIN
```

NOTE:

The keywords PRIMARY and SECONDARY specify the primary and secondary screens on a dual screen UTS 40 workstation. This is an optional parameter; the default designates a single screen workstation. If the workstation is designated as a dual screen workstation, two TERM macroinstructions must be specified, one for each of the screens.

6. SESSION This macroinstruction can't be used with remote workstations.

5.6. COMMUNICATIONS PHYSICAL INTERFACE (CPI) USER GUIDELINES

As a communications physical interface (CPI) user, you are totally responsible for all single line communications adapter (SLCA) initialization and control procedures. Therefore, you must make some changes to your CPI user programs when you migrate to the model 8. The most significant changes are summarized here:

- The communications physical input/output control program (CPIOCP) packet size has been increased to 14 full words (including sense bytes for the SLCA).
- The port control word must specify the character detect and character interpretation tables as table 1 (bytes 2 and 3, bit 1=0).
- The SLCA buffer size must be specified in the TN#PBLTH field of the CPIOCP at line request time. The valid range is 32–256 in decimal. If buffer toggling is required, the line buffer size must be an even multiple of the SLCA buffer size.
- To set asynchronous line speeds for the model 8 single line communications adapter port control word, see Table 5–2.
- The channel number and SLCA number must be set in the CPIOCP for every request, including network request. The channel number must be $0D_{16}$ or $0F_{16}$; the SLCA number must be in the range 01_{16} to $0F_{16}$.
- To get the data in the SLCA buffer when a CPIOCP input timeout error occurs, specify the line procedure timer value in the port control word. (See the current version of the ICAM communications physical interface (CPI) user guide, UP-8945.)
- The turn-off command does not provide an immediate status. The I/O completion interrupt must be awaited.

- Data chain users must set up the first two CPIOCP packets (including the first data chain address) prior to the first CCR call. Failure to comply with this requirement causes a system error.

- Invalid packet chaining links cause system errors.

- If buffer toggling is required, TN#PESO (error schedule only) must not be specified in the TN#PFLGS field of the CPIOCP. If specified, return of the CPIOCP buffer completion interrupt is prevented.

- If buffer toggling is required, all CPIOCPs must specify continuation (TN#PFS and TN#PFL=0) after the first CPIOCP.

Table 5–2. Port Control Word Settings for Model 8 Asynchronous Line Speeds

Bits				Asynchronous Speed (bits per second)
4	5	6	7	
0	0	0	0	Not used (invalid*)
0	0	0	1	50
0	0	1	0	75
0	0	1	1	110
0	1	0	0	134
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	900
1	0	0	1	1200
1	0	1	0	1800
1	0	1	1	2400
1	1	0	0	3600
1	1	0	1	4800
1	1	1	0	7200
1	1	1	1	9600

*Results in program alert sense and unit check status

For network generation:

- If your line handler uses half-duplex protocol, you must specify half-duplex in the COMMCT LINE macroinstruction and CACH statement. All ICAM line handlers use half-duplex except NTR, UDLC–ABM, and PDN level 2X.25.

- If your line handler uses full-duplex protocol, you must also specify full-duplex in the COMMCT.

- If necessary, you can use full-duplex modems and lines with half-duplex line handler protocols.

6. IMS Migration

6.1. GENERAL

As you know, you can generate two types of information management systems (IMS) on your Series 90 system:

1. Single-thread IMS

   Only one action is processed at a time, but actions from different transactions may be interspersed.

2. Multithread IMS

   Actions are processed concurrently for different transactions.

Your existing Series 90 IMS systems can run without change on the model 8. But, if you want to incorporate new software features into your IMS action programs, you must:

- Convert your Series 90 single-thread IMS systems to multithread
- Convert action programs that use basic data management (BDM) to use consolidated data management (CDM)

Whenever possible, we recommend that you make these changes to you can use all the features offered by the model 8 and by release 8.2 IMS.

In addition, if you convert to CDM, you must convert all your IMS systems at the same time. IMS does not support mixed mode applications. Therefore, all your IMS systems must run under one type of data management – either BDM or CDM, but not both.

6.2. CONVERSION TO MULTITHREAD IMS

To convert from single-thread to multithread IMS you must:

- change IMS configuration parameters not supported by the model 8; and
- change the associated job control streams.

In the following subsections, we show you which IMS configuration parameters you must change when moving to a multithread environment. We also provide a sample job control stream so you can see how to execute online IMS on the model 8. However, remember that when you upgrade to multithread IMS, you should review the overall logic and design of your action programs. In this way, you can take advantage of multithread to improve the efficiency of your IMS system.

6.2.1. IMS Configuration Parameters

Table 6–1 shows which IMS configuration parameters are supported for OS/3 releases 6.1.2 forward. To check if a parameter is supported on the model 8, read across the table to the column under release 8.2. An X in those columns indicates that the parameter is supported for IMS single-thread or multithread. If a parameter is not supported for model 8 release 8.2, you must remove it from your configuration when you migrate to the model 8. For a complete listing of parameter definitions and values, refer to the IMS system support functions user guide, UP-8364 (current version).

Table 6–1. IMS Configuration Parameters

Section	Parameter	Release 6.1.2		Release 7.0.1/7.1		Release 8.0		Release 8.2	
		Single-thread	Multi-thread	Single-thread	Multi-thread	Single-thread	Multi-thread	Single-thread	Multi-thread
NETWORK	CUP			X	X	X	X	X	X
	KATAKANA					X	X	X	X
GENERAL	INBUFSIZ	X	X	X	X	X	X	X	X
	DDPBUF						X		X
	DDPSESS						X		X
	TRANLEN						X		X
OPTIONS	BASIC	X		X					
	DMS	X		X	X	X	X	X	X
	FASTLOAD			X	X	X	X	X	X
	INTLIST	X		X		X	X	X	X
	MSGCLR					X	X	X	X
	MSGPOS					X	X	X	X
	OPCOM		X		X	X	X	X	X
	RESFMT					X	X	X	X
	RECLOCK	X		X		X		X	
	SFS			X		X	X	X	X
	STATS					X	X	X	X
FILE	FILETYPE	X	X	X	X	X	X	X	X
	CAFILE	X	X	X	X	X	X	X	X
	CUPDATE	X	X	X	X	X	X	X	X
	DELETP					X	X	X	X
	PKEY					X	X	X	X
	DTF	X	X	X	X	X	X	X	X
	RIB			X		X	X	X	X
TRANSACT	LOCAP						X		X
	UNDEF					X	X	X	X
ACTION	FCCEDIT			X	X	X	X	X	X
LANGUAGE	lex-name					X	X	X	X
	lang-elem					X	X	X	X
LOCAP	locap						X		X
	RCHAR						X		X
DRCRDMGT	RESIDE	X		X		X		X	

6.2.2. IMS Job Control

The following job control stream represents a typical execution of online IMS. We've provided explanations at the end of the control stream for all numbered items. For more information on IMS job control, refer to the OS/3 IMS system support functions user guide, UP-8364 (current version).

```
      // JOB IMS,,min,,tasks
      // OPTION SYSDUMP
      // DVC 20  // LFD PRNTR
      // DVC lun  // VOL vsn
      // LBL file-id-1  // LFD data-file-name
                •

                •
      // DVC lun  // VOL vsn
      // LBL file-id-n  // LFD data-file-name
① [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD NAMEREC
  [  // DVC lun  // VOL vsn
② [  // LBL file-id  // LFD AUDFILE
  [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD CONDATA
③ [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD STATFIL,,INIT
  [  // DVC lun  // VOL vsn
④ [  // LBL file-id  // LFD TCIDTF
  [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD disk-queue
     // DVC lun  // VOL vsn
     // LBL file-id  // LFD TRCFILE,,EXTEND
⑤ [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD load-lib-name
⑥ [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD LOAD
⑦ [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD LDPFILE
  [  // DVC lun  // VOL vsn
⑧ [  // LBL file-id  // LFD TC01FMTF
  [  // DVC lun  // VOL vsn
  [  // LBL file-id  // LFD TC02FMTF
     // EXEC load-module-name,load-lib-name,priority
     // PARAM parameter-statement
           •

           •
     /&
     // FIN
```

Explanation:

① Allocate NAMEREC file as ISAM in BDM mode and as MIRAM in CDM mode.

② AUDFILE and CONDATA files are used for multithread IMS; single-thread IMS uses one AUDCONF file.

③ Device assignment set for STATFIL.

④ For global network, assign ICAM files in GUST job control stream.

⑤ IMS execution load library.

⑥ IMS program load library; LFD must be LOAD when using fastloader.

⑦ Provide LDPFILE when using fastload feature.

⑧ Screen format files; must be TC01FMTF and TC02FMTF on the model 8.

6.3. CONVERSION OF IMS PROGRAMS TO CDM

To convert your IMS systems from BDM to CDM, you must:

1. Accept the default CDM=YES in IMSCONF.
2. Change data management parameters in the FILE section of the IMS configurator to support CDM.
3. Convert your data files to MIRAM format.
4. Recompile your action programs.

Subsection 6.2.1 describes the required changes to IMS configuration parameters; conversion of data files to MIRAM is discussed in Section 2. The following subsections describe the changes you must make to IMSCONF and to your IMS action programs.

6.3.1. IMSCONF

The IMSCONF job control procedure (jproc) generates and executes control streams to perform system generation for single-thread and multithread IMS systems. Use the following IMSCONF parameter to select the data management mode your IMS system uses:

$$\text{CDM=}\left\{\begin{array}{l}\text{YES}\\ \text{NO}\end{array}\right\}$$

To generate an IMS system using CDM, accept the default CDM=YES. This parameter replaces the CDI parameter used on Series 90 release 7.1. Note that you must specify CDM=NO, to run IMS in BDM mode; if you specify this parameter, you must also specify DMGTMODE=MIXED when you generate your supervisor during SYSGEN.

An IMSCONF feature, the internal statistical file (STATFIL), records data during online processing. STATFIL is a sequential unkeyed MIRAM file with variable-length records. There are two ways you can allocate STATFIL:

- specify the STATFIL parameter in IMSCONF; or
- specify a user file label in the IMSFIL parameter in IMSCONF.

If you don't want to allocate STATFIL, specify NO in the IMSCONF INIT parameter.

Table 6–2 lists new IMSCONF parameters for releases 6.1.2 forward. An X indicates that the parameter is supported for that release.

Table 6–2. New IMSCONF Parameters

IMSCONF Parameter	Release 6.1.2	Release 7.0.1/7.1	Release 8.0	Release 8.2
CDI	N/A	X	N/A	N/A
CDM	N/A	N/A	X	X
STATFIL	N/A	N/A	X	X
INIT= cyl-statfil	N/A	N/A	X	X
ZCNF=ST	X	X	X	X
ZCNF=MT	X	X	X	X

6.3.2. Action Programs

To migrate your COBOL, RPG II, or BAL action programs, you must recompile those programs on the model 8. The only exception to this is ANSI 1968 COBOL action programs. You must convert your ANSI 1968 COBOL action programs to ANSI 1974 COBOL and then recompile those programs on the model 8. Section 3 provides details on converting ANSI 1968 COBOL to ANSI 1974 COBOL.

7. Software Products

7.1. GENERAL

In this section we'll tell you how to convert the following software products for use on the model 8:

- Screen formats
- BEM EDT procedures
- SORT/MERGE and SORT3 job control streams
- Data base management system (DMS 90)

7.2. SCREEN FORMAT MIGRATION

You can move your existing Series 90 screen format libraries intact to the model 8. Series 90 screen formats created on release 8.0 are fully compatible between systems and can be run on the model 8 without change. If you are migrating from Series 90 release 7.0.1 or 7.1, you can also use your screens on your new system. But when you use release 7.0.1 or 7.1 screens, the model 8 screen format coordinator must convert each screen each time the screen is read into main storage. Therefore, we recommend that, whenever possible, you convert your 7.0.1 or 7.1 screens to meet model 8 requirements.

We provide a utility program, the screen format conversion utility (SFCNVR), that converts release 7.0.1 or 7.1 screens so they are compatible with the model 8. SFCNVR takes these screens and reworks them into a format acceptable to the model 8 screen format coordinator. You can run SFCNVR either in batch mode or interactively, simply by entering the following command:

```
RV SFCVR[,,I=Y]
```

where:

```
I=Y
```

Specifies that you are running SFCNVR interactively.

Additional keyword parameters let you specify which files or libraries you want to convert. Of course, you can also specify where the converted output is to be stored. In most cases, you'll probably accept the following defaults when you run SFCNVR:

- Old screen formats are stored in $Y$FMT on your release 7.0.1 or 7.1 SYSRES volume.
- New (converted) screen formats are stored in $Y$FMT on your model 8 SYSRES volume.

Just as a precaution, we recommend that you back up and save your original screen formats before you use SFCNVR. Store these formats in a library until your migration is complete. You'll find that SFCNVR lets you modify a complete format library or individual format files quickly and simply. For more information on this product, see the OS/3 Release 8.2 system release description (SRD).

7.3. BEM MIGRATION

The basic editor monitor (BEM) is not supported on the model 8. Instead, the model 8 provides an interactive services editor (EDT) that runs in either CDM or mixed mode and operates in essentially the same way as BEM EDT. However, there are some differences in the two editors. The following guidelines should help you convert your BEM EDT procedures to EDT with a minimum of difficulty:

- File and record differences:

  1. EDT can access SAT and MIRAM library files, MIRAM data files, spool files, sequential device files, tape, card, and diskette files. BEM EDT accesses only SAT library files.

  2. EDT supports variable-length and fixed-length records. BEM EDT supports only fixed-length records.

- Syntax conventions:

  1. EDT uses a colon (:) as a column and line range separator, while BEM EDT uses a hyphen (-).

  NOTE:

  EDT lets you change the column and line range separator from the preset colon to another symbol via the @SET COLON=range-separator command.

2. EDT indicates the start column of a found search string by an open bracket ([). BEM EDT uses a percent sign (%). EDT uses a closed bracket (]) to indicate the end column of a found search string. BEM EDT has no special symbol for the end column.

3. EDT uses #Gn(x:y) where x:y denotes the column range to define a variable substring. BEM EDT uses #Gn(s,L) where s denotes the starting column and L denotes the length of the substring.

4. EDT uses n(x:y) as the syntax for a line substring; BEM EDT uses n:x–y.

5. The current line number and increment are not local for BEM EDT.

6. EDT (using a free-form scan) recognizes as command lines only those lines whose first nonblank character is a command trigger (@). BEM EDT recognizes the command trigger only in column 1.

7. EDT defaults to the PRINT option of the procedure file that called it, or to NOPRINT if it is called from the main work file. BEM EDT defaults to NOPRINT. EDT allows greater control and flexibility with the PRINT and NOPRINT options on the DO command.

- Function key support:

  1. EDT supports the following function keys:

     F1
     F2
     F15 (EOF)
     F18
     F19

- BEM EDT commands not supported by EDT:

  1. These commands have no equivalent function in EDT:

```
@RSP
@SET PAGE
```

NOTE:

Although EDT does not support RSP scanning of the spool file, you can use the interactive services DI SPL instead. You can also read spool files using EDT; however, you must know which spool file is to be retrieved before you can use this command.

2. The model 8 has equivalent system commands that replace these BEM EDT commands:

```
@UPPER    @HELP
@LOWER    @TYPE
```

3. The EDT keyword KKEY has the same function as the BEM EDT @DESEQ command, but is more thorough in syntax checking.

- BEM features not provided by interactive services:

  1. Idle terminal timeout and logoff

  2. Ability to limit command access based on user-id

  3. TTY support

For further information on EDT, see the OS/3 general editor (EDT) user guide/programmer reference, UP-8828 (current version).

7.4. SORTING

The model 8 supports the sort/merge and SORT 3 programs as follows:

- Sort/Merge

  All Series 90 sort/merge job control streams are fully compatible with the model 8. In addition, the model 8 sort/merge program supports MIRAM files in both BDM and CDM mode.

- SORT3

  All Series 90 SORT3 job control streams are fully compatible with the model 8. SORT3 supports MIRAM files in both BDM and CDM mode. In a mixed mode environment, SORT3 produces MIRAM files as output regardless of input file type.

7.5. DATA BASE MANAGEMENT SYSTEM (DMS 90)

Device media control language (DMCL) modules generated for Series 90 release 7.1 forward are compatible with release 8.2 and can be moved intact to the model 8. However, DMCL modules generated prior to release 7.1 are incompatible. You must recompile these modules before using them on model 8 release 8.2.

Index