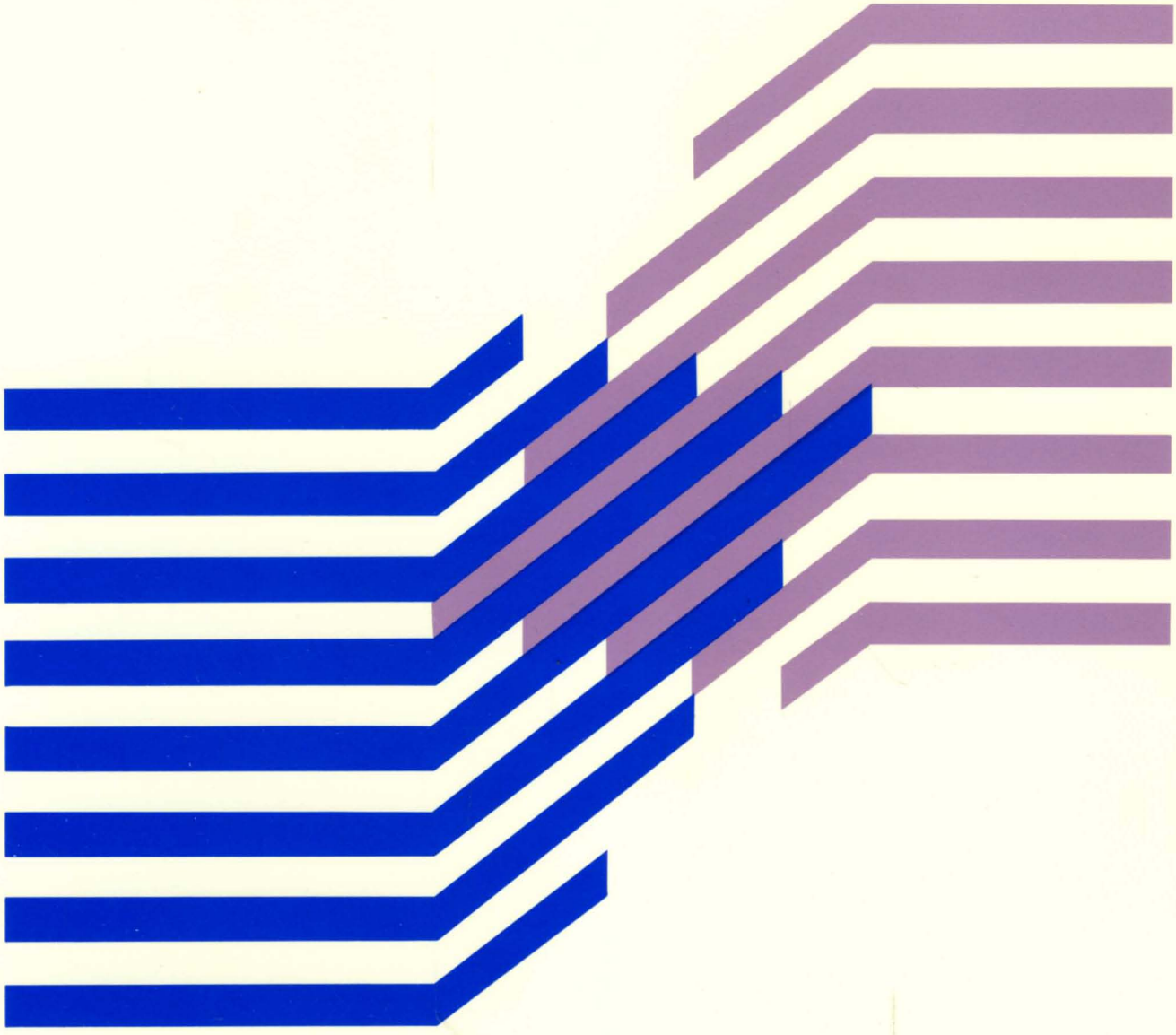
GC28-1817-1

MVS/ESA
MVS Configuration Program
Guide and Reference

MVS/System Product:
JES2 Version 3
JES3 Version 3



GC28-1817-1

MVS/ESA
MVS Configuration Program
Guide and Reference

MVS/System Product:
JES2 Version 3
JES3 Version 3

Second Edition (December, 1989)

This is a major revision of, and obsoletes, GC28-1817-0. See the Summary of Changes following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Version 3 of MVS/System Product program number 5685-001 or 5685-002, and to all subsequent releases, until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to makes these available in all countries in which IBM operates. References to IBM products in this document do not imply that functionally equivalent products may be used. The security certification of the trusted computing base that includes the products discussed herein covers certain IBM products. Please contact the manufacturer of any product you may consider to be functionally equivalent for information on the product's security classification. This statement does not expressly or implicitly waive any intellectual property right IBM may hold in any product mentioned herein.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 950, Poughkeepsie, New York 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PROGRAMMING INTERFACES

This book does NOT contain any programming interface information.

About This Book

The MVS configuration program is used by installations that have installed MVS/ESA™ System Product Version 3 Release 1.0 (MVS/ESA SP3.1.0) or a later release. These installations use the MVS configuration program to:

- Define new I/O configurations or eligible device tables
- Replace existing I/O configurations or eligible device tables
- Define the consoles that the nucleus initialization program (NIP) can use
- Migrate I/O configurations or eligible device tables that were previously defined through the SYSGEN process so they can be used on the MVS/ESA SP3.1.0 or a subsequent release.

This book explains how to use the MVS configuration program to do these tasks.

Notes:

1. Although the MVS configuration program is not used to delete I/O configurations or eligible device tables, this book also explains how to do those tasks.
2. At various places this book uses the acronym MVSCP to refer to the MVS configuration program.

Trademarks

The following are trademarks of International Business Machines Corporation.

- MVS/ESA™
- MVS/SP™
- MVS/XA™

Who This Book Is For

This book is intended for anyone using the MVS configuration program, usually the installation's systems programmers.

How This Book Is Organized

This book is divided into the following chapters and appendixes:

- **Chapter 1, "Introduction"** introduces the MVS configuration program and compares its processing to that of the SYSGEN process. First the chapter explains how the SYSGEN process created an I/O configuration. This is followed by an explanation of how the MVS configuration program creates an I/O configuration. This comparison will help first time users of the MVS configuration program understand the difference between the SYSGEN process and the MVS configuration program process.

  Finally, the chapter introduces the MVS configuration program external interfaces and explains the purpose of each.

- **Chapter 2, "Concepts, Definitions, and Relationships"** is intended for anyone who is unfamiliar with I/O concepts and definitions. Terms such as device type and model number and concepts such as generic device types, esoteric device groups, and the device preference table are explained.
- **Chapter 3, "Planning an I/O Configuration"** provides a checklist of items you should consider when you plan an I/O configuration.
- **Chapter 4, "Managing I/O Configurations"** contains the step-by-step procedures that explain how to do the following major tasks:
  - Migrate an I/O configuration previously defined through the SYSGEN process
  - Define a new I/O configuration
  - Update an I/O configuration
  - Delete an I/O configuration or an eligible device table

  This chapter also includes a decision table to help you select the correct procedure to use.
- **Chapter 5, "MVS Configuration Program Input Statement Reference"** explains how to code the MVS configuration program input statements. These are the statements that define your I/O configuration.
- **Chapter 6, "Defining I/O Devices"** explains, in detail, how to code an IODEVICE statement to define specific I/O devices. Each explanation identifies required parameters, optional parameters, and device features that you can specify.
- **Chapter 7, "Defining Eligible Device Tables and NIP Consoles"** provides detailed instructions for defining an eligible device table and for identifying the consoles that NIP will be allowed to use.
- **Chapter 8, "I/O Configuration Management Tasks"** describes how to do the subtasks that you must do as part of the major tasks that are described in Chapter 4, "Managing I/O Configurations."
- **Chapter 9, "Reading MVS Configuration Program Reports"** explains how to read the MVS configuration program reports.
- **Chapter 10, "Debugging, Messages, and Codes"** explains what to do in case of an error. Included is:
  - An explanation of the output you will receive if there is an input stream error and an explanation of the output you will receive if there is a logic error in the MVS configuration program
  - Instructions on how to read the MVS configuration program messages
  - The MVS configuration program messages and reason codes
- **Appendix A, "Generic Device Types"** lists the IBM-provided generic device types.
- **Appendix B, "IBM Provided Device Preference Table"** identifies the device order that is defined by the IBM-provided device preference table.
- **Appendix C, "Determining 3350P and 3351P Device Number Assignments"** explains how to calculate the value of the device numbers that the MVS configuration program automatically assigns to the 3350P and the 3351P devices.

- **Appendix D, "MVS Configuration Program Example"** contains an example of an MVS configuration program input stream.
- **Appendix E, "Eligible Device Table Verification Service"** explains how to use program IEFEB400 to validate an eligible device table (EDT).

How to Use This Book

If this is the first time you have used this book:

- Read Chapter 1, "Introduction" on page 1-1.
- If you are unfamiliar with I/O concepts and definitions, read Chapter 2, "Concepts, Definitions, and Relationships" on page 2-1.
- Read Chapter 3, "Planning an I/O Configuration" on page 3-1.
- Read Chapter 4, "Managing I/O Configurations" on page 4-1 up to and including "Selecting the Correct Procedure" on page 4-3.

You should also read the introductory section of each chapter. Those sections can help you become familiar with where information is in the book.

Once you are familiar with the book and are ready to use the MVS configuration program, turn to the decision table in "Selecting the Correct Procedure" on page 4-3. That table will help you select the correct procedure for the task that you wish to do.

After you have selected the procedure, look it over to become familiar with it. To do the procedure, start with step 1 and follow the procedure step-by-step. If there is a step that you do not need to do, go to the next sequential step. If a step contains insufficient information for you to do the step, read the material referenced by the step.

If you have problems understanding the format or the organization of the input stream, look at the example in Appendix D, "MVS Configuration Program Example" on page D-1. This example might provide you with the information that you need to proceed.

If the MVS configuration program issues error messages or abnormally terminates, follow the instructions given in the message documentation found in Chapter 10, "Debugging, Messages, and Codes" on page 10-1.

Related Information

Where necessary, this book references information in other books, using shortened versions of the book title. The following table shows the shortened titles, complete titles, and order numbers of the books that you might need while you are using this book.

Short Title	Publication Title	Order Number
Component Diagnosis: MVSCP	*MVS/ESA Component Diagnosis: MVS Configuration Program*	LY28-1852
Data Areas	*MVS/ESA Diagnosis: Data Areas, Volumes 1 to 5*	LY28-1043 to 1047
Initialization and Tuning	*MVS/ESA System Programming Library: Initialization and Tuning*	GC28-1828
IOCP User's Guide and Reference	*IBM ES/3090 Processor Complex Input/Output Configuration Program User's Guide and Reference*	SC38-0066
JCL Reference	*MVS/ESA JCL Reference*	GC28-1829
JCL Users Guide	*MVS/ESA JCL Users Guide*	GC28-1830
JES3 Initialization and Tuning	*MVS/ESA System Programming Library: JES3 Initialization and Tuning*	SC23-0073
MVS/XA Message Library: System Codes	*MVS/Extended Architecture Message Library: System Codes*	GC28-1157
System Codes	*MVS/ESA Message Library: System Codes*	GC28-1815
System Commands	*MVS/ESA Operations: System Commands*	GC28-1826
System Generation	*MVS/ESA System Generation*	GC28-1825
System Messages	*MVS/ESA Message Library: System Messages, Volumes I and II*	GC28-1812 and GC28-1813
System Modifications	*MVS/ESA System Programming Library: System Modifications*	GC28-1831
Utilities	*MVS/ESA Data Administration: Utilities*	SC26-4516
RACF Security Administrator's Guide	*RACF Security Administrator's Guide*	SC28-1340
	Assembler H Version 2 Application Programmer's Guide	SC26-4036
	Assembler H Version 2 Application Programmer's Language Reference	GC26-4037
	Introduction to the IBM 3250 Graphics Display System	GA33-3035
	OS/VS Graphic Programming Support (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System	GC27-6971
	OS/VS Mass Storage Control Table Create	GC35-0013
	OS/VS Mass Storage System (MSS) Installation Planning and Table Create	GC35-0028
	OS/VS Message Library: VS2 System Codes	GC28-1008
	3350/3344 Installation and Conversion Guide	GC20-1780

Contents

Figures

Summary of Changes

Summary of Changes
for GC28-1817-1
MVS/System Product Version 3 Release 1.3

New Information: Chapter 6, "Defining I/O Devices" explains how you can provide additional device security through RACF.

Changed Information: There are miscellaneous technical and editorial changes throughout.

Summary of Changes
for GC28-1817-0
MVS/System Product Version 3 Release 1.0

This book contains information previously presented in *MVS/Extended Architecture MVS Configuration Program Guide and Reference*, GC28-1335-0. The following summarizes the changes to that information.

New Information: Support for the 3180 display station and several new index entries have been added.

Changed Information: Minor maintenance changes, corrections, and editorial changes have been made.

Chapter 1. Introduction

Before MVS/System Product Version 2 Release 2.0 (MVS/SP™2.2.0), installations used the SYSGEN process to define I/O configurations to MVS. When this book refers to defining an I/O configuration to MVS, it means:

- Identifying the available I/O devices.
- Defining esoteric device groups and naming each group.
- Defining eligible device tables (EDTs) and assigning esoteric device groups to these tables.
- Identifying devices that were eligible for virtual I/O (VIO).
- Defining changes to the IBM-supplied device preference table.

Installations that install MVS/ESA, or subsequent releases, must use the MVS Configuration Program to perform these functions. In addition, these installations will also use the MVS configuration program to:

- Identify I/O devices that the Nucleus Initialization Program (NIP) may use as consoles.
- Request I/O configuration data for the JES3 initialization stream checker. This data will help JES3 identify inconsistencies between the MVS configuration definition and the JES3 configuration definition.

Locating Information in this Chapter

This chapter is intended only to give you an overview look at the MVS configuration program and to introduce MVS configuration program terminology. It is not intended to provide the detailed information that you need to use the MVS configuration program. That is the job of the other chapters.

This chapter contains the following introductory topics:

SYSGEN Overview

During a SYSGEN, input statements describe the I/O configuration. Collectively these statements are called the *I/O configuration definition*. The SYSGEN process uses the I/O configuration definition as input to create the control blocks that provide MVS with an internal representation of the I/O configuration. These control blocks are called the *MVS I/O configuration data*.

The SYSGEN process combines the MVS I/O configuration data with the MVS nucleus to form a member (IEANUC0x) of the SYS1.NUCLEUS data set. The system loads this member, and the member becomes the MVS nucleus.

Multiple I/O Configurations

When an installation defines multiple I/O configurations, the SYSGEN process combines the MVS I/O configuration data for each configuration with a separate but identical copy of the MVS nucleus. Combining the MVS I/O configuration data with the nucleus creates a separate IEANUC0x member for each I/O configuration. At IPL, the operator specifies which IEANUC0x member is to be used as the MVS nucleus. In this way the operator selects a specific I/O configuration.

Figure 1-1 shows how SYSGEN supports multiple I/O configurations; shown as I/O configurations one, two, and three, with separate copies of the MVS nucleus to form three IEANUC0x load modules.

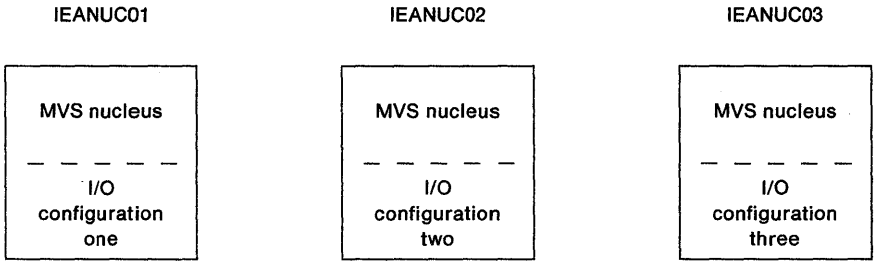


Figure 1-1. Multiple I/O Configuration Support before MVS/SP2.2.0

The way that SYSGEN supports multiple I/O configurations requires, for each configuration, a separate copy of the MVS nucleus. This method of support causes extra maintenance: user modifications that are applied to the nucleus have to be applied to each copy of the nucleus. The same is also true of IBM-provided modifications. In addition, each time an installation changes an existing I/O configuration, it might have to reapply any modifications previously made to the related copy of the nucleus.

MVS Configuration Program Overview

The MVS configuration program replaces the I/O definition functions (IOGEN and EDTGEN) previously available through SYSGEN. The MVS configuration program, in combination with other program enhancements, enables one copy of the MVS nucleus to support multiple I/O configurations. Also, with the MVS configuration program, modifications to the nucleus do not have to be reapplied after an installation updates an existing I/O configuration.

Note: In addition to defining each I/O configuration to MVS, you or another systems programmer must also use the I/O configuration program (IOCP) to define the configuration to the channel subsystem. You can use the same input stream to define the configuration to both the MVS configuration program and IOCP. For information on how to use the IOCP, see the *IOCP User's Guide and Reference*.

Figure 1-2 on page 1-4 shows the data sets and data flow for the I/O definition process. The figure shows inputs to the process and outputs that the process produces. Following the figure is a brief description of each major data set used in the process.

Although the figure shows the MVS configuration program and the IOCP running at the same time, you can actually run these programs at separate times independent of each other. They are shown together only to remind you that they share a common input stream.

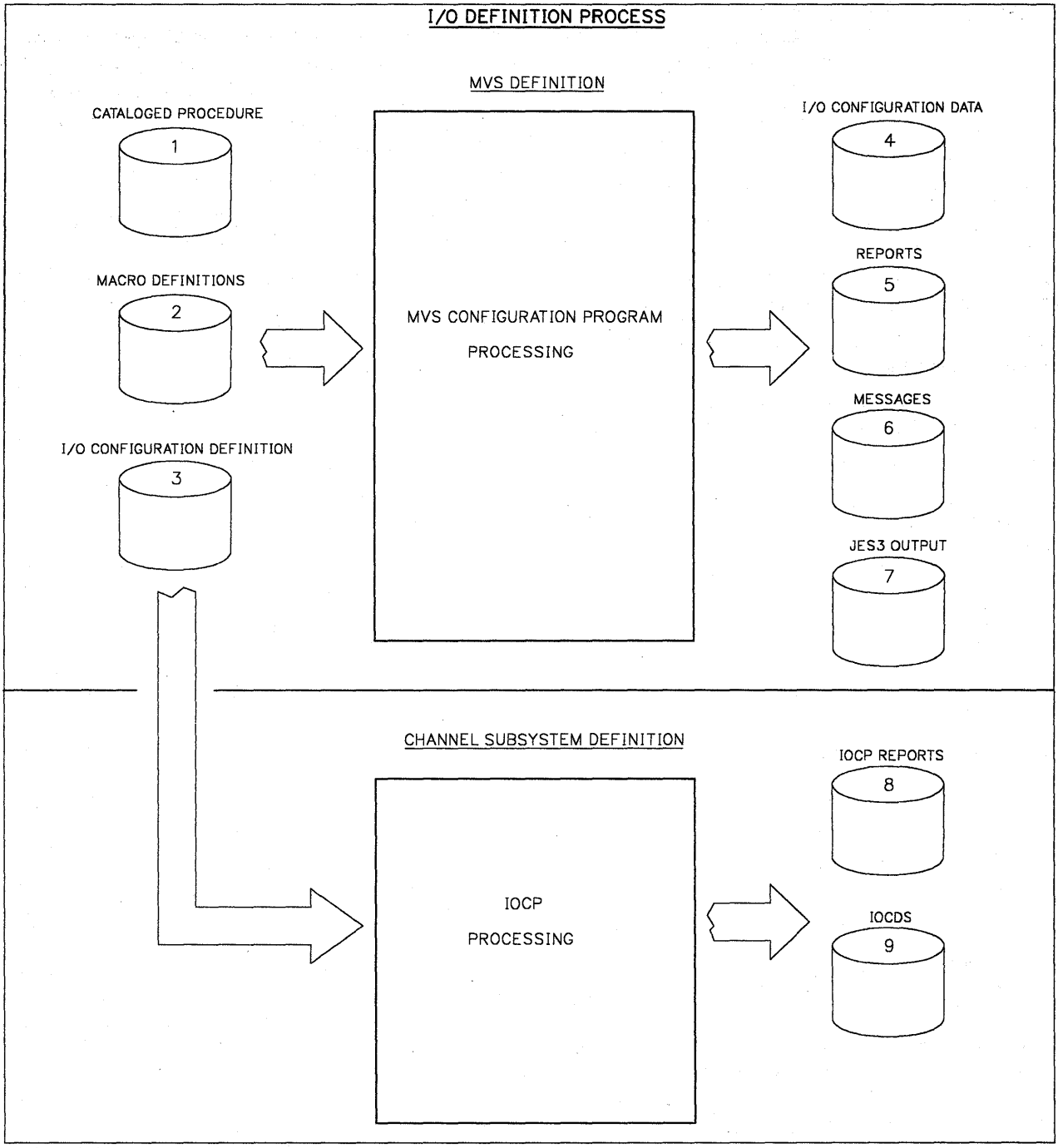


Figure 1-2. MVS and Channel Subsystem I/O Definition Process

Data Set Descriptions

The data sets shown in the previous figure are described below. In the description, DSN and DD Name correspond to the data set name and the DD name, respectively, specified in either the IBM-provided catalog procedure or the IBM-provided sample JCL statements. The catalog procedure and the sample JCL are shown in "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4.

Data set	*DSN*	*DD Name*
(1) ***Invocation JCL*** The cataloged procedure in SYS1.PROCLIB or the sample JCL in SYS1.SAMPLIB used to invoke the MVS configuration program.	SYS1.PROCLIB SYS1.SAMPLIB	none
(2) ***Macro definitions*** Assembler macro definitions that the Assembler uses to parse MVS configuration program input statements.	SYS1.MACLIB	SYSLIB
(3) ***I/O configuration definition*** The MVS configuration program input statements and the IOCP input statements that describe an I/O configuration. You code these statements.	Not applicable	SYSIN
(4) ***I/O configuration data*** A partitioned data set (PDS) or partitioned data set extended (PDSE) that contains the I/O configuration that the MVS configuration program creates. The MVS configuration program stores this data into members, IEANCTxx, IEFEDTxx, IOSIITxx, and IOSUCBxx. **Note:** The value xx appended to the member names is an identifier that you define. It distinguishes one configuration from another. The identifier's use is explained in more detail later in the topic "Output Identifiers" on page 1-10.	SYS1.NUCLEUS	SYSLMOD
(5) ***Reports*** MVS configuration program reports: • **I/O supervisor (IOS) report** contains an entry for each device defined in the configuration • **Allocation report** contains eligible device table information • **NIP report** lists the devices that NIP can use as consoles	Not applicable	SYSRPT
(6) ***Messages*** Messages that were issued during the I/O definition process.	Not applicable	assembler messages: SYSASMP loader messages: SYSLOUT linkage editor messages: SYSLKEDP MVSCP messages: SYSPRINT

Data set	*DSN*	*DD Name*
(7) ***JES3 Output*** Data that the MVS configuration program produced for the JES3 initialization stream checker.	Not applicable	SYSJES3
(8) ***IOCP reports*** IOCP reports.	Not applicable	Not applicable
(9) ***IOCDS*** - The I/O configuration data for the channel subsystem	Not applicable	Not applicable

For a functional description of all of the files used by the MVS configuration program, see "Standard DD Name List" on page 8-19.

Note: The MVS configuration program resides in SYS1.LINKLIB.

The remainder of this chapter provides an overview of the MVS configuration program and how you use it to define an I/O configuration to MVS.

MVS Configuration Program Characteristics and Interfaces

The MVS configuration program runs as a problem program on MVS/370, MVS/XA™, or MVS/ESA. Each time you run the MVS configuration program, you can define only one I/O configuration. You can, however, define multiple eligible device tables. If you need to define more than one I/O configuration you must run the MVS configuration program once for each configuration.

Although the MVS configuration program runs on MVS/370, you can use the I/O configurations that it builds on MVS/SP2.2.0 or a subsequent release only.

Note: You should run the MVS configuration program through either the cataloged procedure or JCL as detailed in "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4.

MVS Configuration Program Input Statements

You code input statements to describe an I/O configuration to the MVS configuration program. The MVS configuration program and the IOCP can share a common input stream that contains both MVS configuration program input statements and IOCP input statements. Collectively, these statements are called the I/O configuration definition.

The MVS configuration program accepts, as SYSGEN did, the following input statements:

- **CHPID**
- **CNTLUNIT**
- **ID**
- **IODEVICE**
- **UNITNAME**

In addition, the MVS configuration program accepts three new input statements:

- **IOCONFIG**
- **EDT**
- **NIPCON**

The eight statements listed previously are the *only* input statements that should appear in the input stream.

The following table summarizes the purpose of each statement and identifies which program, the MVS configuration program or IOCP, uses the statements:

Figure 1-3. Input Statement Summary

Statement	Used by	Purpose
CHPID	IOCP	Define channel paths.
CNTLUNIT	IOCP	Define control units
EDT	MVS configuration program	Define an eligible device table, assign it an identifier, and define changes to the device preference table.
ID	IOCP MVS configuration program	Define data that is to appear in the headings of IOCP reports and in the headings of MVS configuration program reports.
IOCONFIG	MVS configuration program	Assign an identifier to an I/O configuration.
IODEVICE	IOCP MVS configuration program	Define an I/O device.
NIPCON	MVS configuration program	Identify devices that NIP will be allowed to use as consoles.
UNITNAME	MVS configuration program	• Define and name an esoteric device group and indicate whether it is eligible for virtual input/output (VIO). • Identify generic device types that you want to make eligible for virtual I/O (VIO).

For a detailed description of the EDT, IOCONFIG, IODEVICE, NIPCON, or UNITNAME statements, see Chapter 5, "MVS Configuration Program Input Statement Reference" on page 5-1. For a detailed description of the CHPID, CNTLUNIT, or ID statements, see the *IOCP User's Guide and Reference.*

IBM-Provided Cataloged Procedure and Sample JCL

IBM provides both a cataloged procedure and sample JCL statements that you can use to invoke the MVS configuration program. The cataloged procedure is in SYS1.PROCLIB(MVSCP); the sample JCL is in SYS1.SAMPLIB(MVSCPXA) and SYS1.SAMPLIB(MVSCP370). Which one you use depends on the MVS release level on which you run the MVS configuration program and the MVS release level on which you intend to use the I/O configuration. For information about using the cataloged procedure or the SAMPLIB JCL, see "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4.

Devices Supported

The MVS configuration program supports many IBM-built devices and several non-IBM devices. For a list of the supported devices, look in the index for the specific device type (for example, 3330 or 3380), or look in the index under the entry *devices supported.*

Note: The MVS configuration program supports all of the devices that the MVS SYSGEN process supported except the 2260.

For each device type that it supports, the MVS configuration program provides a program called a unit information module (UIM). The UIMs process the IODEVICE statements and provide device-dependent data.

If you have a device that the MVS configuration program does not support, you can still use that device. To make such a device available for use, you must either define it as a DUMMY device or provide a UIM to support it.

For more information about defining an unsupported device, see "Defining an Unsupported Device" on page 6-4. For details about coding a UIM, see *System Modifications*.

MVS Configuration Program Modes of Operation

The MVS configuration program has four modes of operation: IODEF, EDT, VALIDATE, and DEBUG. After you select a mode and invoke the MVS configuration program, the MVS configuration program processes the input statements according to the mode selected. During processing, the MVS configuration program invokes the Assembler, the loader, and the linkage editor.

The following table explains the purpose of each MVS configuration program mode.

Figure 1-4 (Page 1 of 2). Overview of the Modes of Operation

Mode	Purpose and MVS Configuration Program Action
IODEF	Used to create a new configuration or to update an existing configuration. • For a new configuration, the MVS configuration program writes new configuration members • For an existing configuration, the MVS configuration program replaces all members of the configuration. • Optionally, the MVS configuration program writes I/O configuration related data for use by JES3. • The MVS configuration program also writes the IOS report, the allocation summary report, and the NIP report. IODEF mode is functionally similar to a SYSGEN IOGEN.
EDT	Used to replace an existing eligible device table (EDT) or to create a new EDT for an existing configuration. • If you update an EDT, the MVS configuration program replaces the updated member of the configuration. • If you create a new EDT, the MVS configuration program writes a new member for the configuration. • Optionally, the MVS configuration program writes I/O configuration related data for use by JES3. • The MVS configuration program also writes the IOS report, the allocation summary report, and the NIP report. EDT mode is functionally similar to an EDTGEN.

Figure 1-4 (Page 2 of 2). Overview of the Modes of Operation

Mode	Purpose and MVS Configuration Program Action
VALIDATE	Used to validity check MVS configuration program input statements *without* creating or replacing a configuration. • The MVS configuration program writes error messages for each error it finds on an input statement. • The MVS configuration program also writes the IOS report, the allocation summary report, and the NIP report. **Notes:** 1. The MVS configuration program also validity checks input statements in both IODEF and EDT modes. 2. There are certain input stream errors that the MVS configuration program detects in IODEF mode but not in VALIDATE mode.
DEBUG	Used to produce additional output when you suspect a problem with the MVS configuration program. • The MVS configuration program validity checks the input statements and builds the configuration but does not write it to the output data set. • The MVS configuration program prints the Assembler text records and the loader output. • The MVS configuration program also writes the IOS report, the full set of allocation reports, and the NIP report.

For a more detailed explanation of when to use each mode, see "Selecting the Mode of Operation" on page 8-15.

MVS Configuration Program Output

The MVS configuration program output depends on the mode that you select and whether there are errors in the input statements. Detectable errors in the input statements cause the MVS configuration program to issue messages and, depending on the severity of the error, can prevent the MVS configuration program from writing the I/O configuration data.

I/O Configuration Members: When the MVS configuration program writes I/O configuration data, it writes it to members of a partitioned data set (PDS/PDSE). The members are named IEANCTxx, IEFEDTxx, IOSIITxx, and IOSUCBxx.

The number of members written depends on the number of eligible device tables defined in the I/O configuration definition. For each I/O configuration, the MVS configuration program writes one each of the following: IEANCTxx, IOSIITxx, and IOSUCBxx. In addition, the MVS configuration program writes one IEFEDTxx member *for each* eligible device table. Unlike SYSGEN, the MVS configuration program *does not* combine the I/O configuration data with the MVS nucleus.

In addition to writing I/O configuration data, the MVS configuration program writes messages, reports, and, optionally, data that can be read by the JES3 initialization stream checker.

Output Identifiers: The value *xx* appended to member names is the identifier that you assign.

- For the IEFEDTxx members, it is one of the following:
  - If you specify an identifier on the EDT statement, it is that identifier.
  - If you omit the identifier or omit the EDT statement, it is the identifier on the IOCONFIG statement.
- For the other members, it is the identifier specified on the IOCONFIG statement.

These identifiers enable MVS and the operator to distinguish one I/O configuration from another and one eligible device table from another.

Output Data Set: Before MVS can use the I/O configuration data, the data must be stored in the SYS1.NUCLEUS data set. Two suggested ways to store the data are:

- Allocate the SYS1.NUCLEUS data set as the MVS configuration program output data set. The MVS configuration program will then write the members to the SYS1.NUCLEUS data set.
- Allocate some other PDS/PDSE as the MVS configuration program output data set. After the MVS configuration program executes, you can use a utility program to copy the members to the SYS1.NUCLEUS data set.

Multiple I/O Configuration Support

Through the MVS configuration program, you can define multiple I/O configurations, and within each configuration, multiple eligible device tables. During the IPL process, the operator selects the nucleus, the I/O configuration, and the eligible device table that are to be used. The operator can make the selections explicitly (by specifying the appropriate identifiers) or implicitly by allowing the selections to default.

The default nucleus is IEANUC01. The default I/O configuration is the configuration that has the identifier 00. The default eligible device table is the table that has the same identifier as the selected I/O configuration.

MVS uses the selected nucleus, the selected I/O configuration, and the selected eligible device table until the next IPL. If, after selecting an I/O configuration and EDT, the operator wants to make a different selection, the operator must re-IPL MVS. For detailed information about how the operator makes these selections, see *System Commands*.

The following example and Figure 1-5 on page 1-11 shows how the MVS/System Product supports multiple I/O configurations with one copy of the MVS nucleus.

- The configuration with an identifier of 00 contains one eligible device table which also has an identifier of 00. The data for this I/O configuration consists of members IEANCT00, IOSIIT00, IOSUCB00, and IEFEDT00.
- The configuration with an identifier of AA contains three eligible device tables with identifiers of 10, 11, and 12. The data for this I/O configuration consists of members IEANCTAA, IOSIITAA, IOSUCBAA, IEFEDT10, IEFEDT11, and IEFEDT12.

If, for example, the operator selects I/O configuration AA, eligible device table 11, and allows the nucleus selection to default, the system loads members IEANUC01, IEANCTAA, IOSIITAA, IOSUCBAA, and IEFEDT11. MVS then uses that nucleus, that I/O configuration, and that eligible device table until the next IPL.

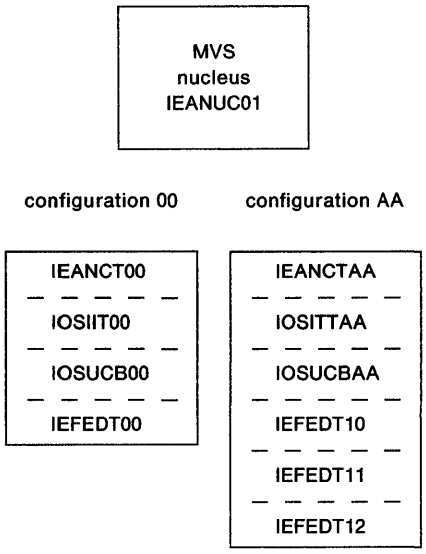


Figure 1-5. MVS/SP Version 3 Multiple I/O Configuration Support

JES3 Initialization Stream Checker Data

During the JES3 initialization process, the JES3 systems programmer must define the I/O configuration to JES3. It is essential that the JES3 definition and the MVS configuration program definition be consistent.

Each time you run the MVS configuration program in either IODEF mode or EDT mode, you can request that it create configuration data for JES3. This data, when read by the JES3 initialization stream checker, permits JES3 to detect inconsistencies between its I/O definition and the MVS configuration program's I/O definition.

You can use the MVS configuration program to create data for the JES3 initialization stream checker regardless of the JES3 release level on which it will be used.

For more information about how JES3 uses this data, see *JES3 Initialization and Tuning*.

Chapter 2. Concepts, Definitions, and Relationships

This chapter is intended primarily for those who are unfamiliar with I/O concepts and definitions. There are several definitions and concepts that you need to understand when reading the rest of this book. This chapter explains those definitions and concepts.

Locating Information in this Chapter

This chapter contains the following major topics:

I/O Device, UCB Relationship

The unit control block (UCB) is the MVS internal representation of an I/O device. The MVS configuration program creates one UCB for each device number indicated on an IODEVICE statement.

I/O Device Terminology

This book uses the following terminology when discussing I/O devices.

Device Type

Each IBM-built I/O device has a type and, in some cases, a model number. The designers of the device determine the device type and model number.

Device type refers to the 4-digit number by which the device is known. For example:

2305
3350
3380

Model Number

The model number, or model, refers to a specific variation of a device type. Model is represented by a dashed number, letter, or both following the device type. For example:

2305-**2**
3333-**11**
3420-**7**

Device Class

MVS groups devices into one of the following device classes:

- Channel-to-channel adapters
- Communication devices
- Direct Access Devices
- Display devices
- Character Readers
- Magnetic Tape Devices
- Unit Record devices

For a list of the devices types that are included in each device class, see Appendix A, "Generic Device Types" on page A-1.

Multiple Exposure Device

This book uses the term *multiple exposure device* to refer to a device that can be *allocated* by a single device number but can be *accessed* by multiple device numbers. Each device number of a multiple exposure device represents one exposure. The device number by which the device is allocated is called the *base exposure*. The other device numbers are called *non-base exposures*.

For some devices each exposure corresponds to a separate device access mechanism. An example of such a device is the 2305 model 2. For other devices that are attached to a control unit that has a cache, each exposure corresponds to a separate logical path to a single access mechanism. Examples of such devices are the 3350P and 3351P.

For a multiple exposure device, the MVS configuration program generates multiple device numbers. You provide the initial device number and a value called the replication factor (the replication factor is also called number-of-units). The MVS configuration program calculates the other device numbers and builds a UCB for each exposure.

The total number of device numbers that the MVS configuration program generates for a particular device depends on two factors:

- The device type
- The value of the replication factor

The values of the generated device numbers also depends on two factors:

- The device type
- The value of the device number that you provide

The multiple exposure devices that the MVS configuration program supports are:

- 2305-2
- 3350 attached to a 3880-11 (called a 3350P)
- 3350 attached to a 3880-21 (called a 3351P)

Generic Device Types

MVS logically groups device types with similar characteristics and assigns the group a generic name. Such a group is called a generic device type. When you refer to a generic device type, you refer to it by its generic name.

MVS, for example, groups the 3330-1, 3330-2, and 3333-1 into a generic device type named 3330. Anytime a program refers to a 3330, MVS interprets it to mean any of the devices in that generic device type.

To request a device allocation, a user can specify a generic device type rather than a specific device number. MVS then allocates a device from the specified generic device type.

For example, if a user coded the following DD statement:

```
//OUTPUT DD UNIT=3330,...
```

MVS would allocate a device from the generic device type named 3330.

Generic device types, permit a user to request an I/O device without specifying a specific device number. For a list of generic names and the device types that correspond to that name, see Appendix A, "Generic Device Types" on page A-1.

Esoteric Device Groups and EDTs

Each time a user submits a job, UNIT= parameters on DD statements or the corresponding dynamic allocation parameters, identify the I/O devices the job requires. Before the job can continue execution, MVS must allocate all of those devices to the job.

If the user requests a *specific device* (specifies a device number on the UNIT= parameter or on the corresponding dynamic allocation parameter), and that device is available, MVS allocates the device to the job. If the device is unavailable, however (it's allocated to another job, for example), the user's job will have to wait until the device becomes available for allocation. By requesting a specific device, the user has limited the choice of devices to one. By limiting the choice of devices, the user increases the probability that the device the job needs will be unavailable when needed.

The user can increase the probability that a device will be available for allocation when needed by requesting a *generic device type*. The increase in probability assumes, of course, that there is more than one device of that generic type available to be allocated.

When a user requests a generic device type, MVS "serializes" on that generic device type. This means that MVS does not allocate devices from that generic device type to other jobs until the user's job allocation request is satisfied. Because MVS makes the entire generic device type unavailable for allocation to other jobs,

there might be a delay in satisfying the other job's allocation requests. Thus, when a user requests a generic device type, it increases the probability that there will be a delay in satisfying allocation requests for other jobs.

Esoteric device groups, eligible device tables (EDTs), and the device preference table provide a means for you to manage these potential allocation performance problems.

What is an Esoteric Device Group?

An *esoteric device group* is an installation-defined and named collection of I/O devices. When a user submits a job, the job can identify a device that it needs by specifying an esoteric device group instead of a specific device number or a generic device type. A job that specifies an esoteric device group is requesting MVS to allocate a device from that group.

When MVS allocates devices to that job, MVS will "serialize" on the esoteric device group until the allocation request is satisfied. Only those devices that belong to the esoteric device group will be unavailable for allocation to other jobs.

Assuming that the esoteric device group identifies more than one device, a device will more likely be available when requested than if the request was for a specific device. Also, if you assume that the esoteric device group does not contain all of the devices from any one generic device type, devices from each generic device type might still be available for allocation to other jobs during the time that MVS is "serialized" on the esoteric device group.

You use the MVS configuration program to define a group, assign it a name, and to identify the I/O devices that you want included in the group. The name you assign to an esoteric device group is called the esoteric name or, in JCL terminology, the user-assigned-group-name. To request allocation of a device from an esoteric device group, a user specifies the esoteric name on the UNIT= parameter of a JCL DD statement.

An esoteric device group can include devices of different generic device types. With one exception, however, you should not define a group that includes devices from more than one device class. For example, a group that includes unit record devices must include *only* unit record devices. The exception: You can define one or more groups that contain both DASD and magnetic tape devices.

What is the Device Preference Table?

An esoteric device group can include devices of more than one generic device type. When MVS attempts to allocate a device from an esoteric device group, MVS needs to be told in what order to attempt the allocation. The device preference table defines that order.

IBM provides a predefined device preference table. The table that IBM provides is defined such that MVS attempts to allocate the fastest device available to each allocation request. For each EDT that you define, you can setup the device preference table in any order you prefer.

For example, assume you have defined an esoteric device group that includes only generic device types 2305-2, 3330, and 3380. Also assume that you have not changed the order of the device preference table. When MVS attempts to allocate a

device from this group, the device preference table tells MVS to attempt allocation in the following order: 2305-2, 3380, and 3330.

The IBM-provided device preference table is shown in Appendix B, "IBM Provided Device Preference Table" on page B-1.

Device Preference Table's Affect on UNIT=AFF Request

When a job requests unit affinity, the sequence of device types as they appear in the device preference table can affect allocation. The sequence affects allocation whenever a DD requests unit affinity to another DD that has requested allocation to an esoteric device group and both DDs specify a status of NEW. When this situation occurs, MVS allocates both DDs to the same physical unit. That unit:

- Will belong to the specified esoteric device group
- Will be the first device type from that esoteric device group to appear in the device preference table

The following example shows how the device preference table affects a UNIT=AFF specification.

Example

Conditions

- Your installation has previously defined an esoteric device group and named it TAPE. Device types 3400-3, 3400-5, and 3480 belong to the group named TAPE.
- Your installation is using the IBM-provided device preference table. In that table affected device types appear in the following order:

  3480
  3400-5
  3400-3

- Assume the following JCL statements:

```
//JOB1  JOB  .....
//STEP1 EXEC .....
//DD1   DD   DISP=NEW,UNIT=TAPE,.....
//DD2   DD   DISP=NEW,UNIT=AFF=DD1,.....
      .
      .
//
```

Explanation

- DD1 requests MVS to allocate a device from the esoteric device group TAPE.
- DD2 requests affinity to DD1.
- MVS allocates DD1 and DD2 to the same physical unit. That unit will be a 3480.

What is an Eligible Device Table (EDT)?

An EDT is an installation-defined and named representation of the devices that are eligible for allocation. The EDT also defines the esoteric and generic relationships of these devices.

During each IPL, the installation identifies the EDT that MVS is to use. Jobs that execute after the IPL can request device allocations from any of the esoteric device groups assigned to the selected EDT.

You use the MVS configuration program to define EDTs, name the EDTs (the name is actually a two-character identifier), and identify the esoteric device groups that each EDT is to define. For each I/O configuration you must define at least one EDT.

What is a Virtual I/O (VIO) Eligible Device?

VIO refers to data set allocations that exist in paging storage only. A real device is not used unless the data set must be paged out. If the data set must be paged out, MVS writes it to a paging device. Programs that use VIO data sets access them just as if the data sets were allocated to a real I/O device.

Users can allocate only temporary data sets to VIO. To request a VIO allocation, the UNIT= parameter on the user's DD statement must specify an esoteric device group or generic device type that is eligible for VIO. The books *JCL Reference* and *JCL Users Guide* explain, in detail, how to make the request. Those books also explain the conditions that must be met before MVS satisfies a VIO allocation request.

You use the MVS configuration program to identify the esoteric device groups or the generic device types that you want to be eligible for VIO allocation. MVS will allocate only DASD for VIO. Therefore, any group that you identify must contain at least one DASD.

What is the NIP Console Table?

The NIP console table identifies, to MVS, the devices that you will allow NIP to use as consoles. You identify these devices to the MVS configuration program by coding one or more NIPCON statements. You must do this for each I/O configuration that you migrate and for each new configuration that you define. MVS attempts to use the devices in the same order that you define them. For more detailed information about defining the NIP console table, see "Defining NIP Consoles" on page 7-9.

Chapter 3. Planning an I/O Configuration

Before you can use the MVS configuration program to define an I/O configuration, you must first plan that configuration. The planning activities include the following:

- Identify the I/O devices that you will include in the configuration.
- Decide how many esoteric device groups to define, identify the I/O devices that you will include in each group, select a name for each group, and identify those groups you want to make eligible for VIO.
- Identify any generic device types that you want to make eligible for VIO.
- Decide how many eligible device tables (EDTs) to define, identify the esoteric device groups that you will include in each EDT, and select an identifier for each EDT.
- Define any changes you wish to make to the device preference table for each EDT.
- Identify the I/O devices that you will allow MVS to use as NIP consoles.

When you complete the planning step, your plan should contain, at a minimum, the following information:

- **Identifier for the I/O Configuration**
- **For each I/O device**
  - Device type and model number (where applicable)
  - Type and model number of the control unit to which each device attaches
  - List of device features, if any
  - Device number you want assigned to device
- **For each esoteric device group**
  - List of devices you will assign to the group
  - Name you will assign to group
  - Whether the group is eligible for VIO
- **For each EDT**
  - ID you will assign
  - Names of esoteric device groups you will assign to EDT
  - Device preference table modifications, if any
- **For NIP consoles**
  - List of devices that you want eligible for use as NIP consoles

After completing the configuration plan, you are ready to define the I/O configuration. Defining the configuration means coding the configuration on input statements and executing the MVS configuration program and possibly the IOCP. When you finish, the internal representation of the I/O configuration will be available on data sets and will be ready for use by the system.

I/O Configuration Planning Checklist

The following checklist identifies planning decisions that you must make and provides references to information that will assist you in making the decisions. After you have made all of these planning decisions you will have a plan for an I/O configuration. Step-by-step procedures in Chapter 4, "Managing I/O Configurations" on page 4-1 will help you define the configuration that your plan describes.

Planning Activities

Planning Decisions You Must Make:	References
Decide which I/O devices to include in the configuration.	None.
Determine if your configuration will include any unsupported devices	"Determining if The MVS Configuration Program Supports a Device" on page 6-3.
Decide how to support any unsupported devices.	"Defining an Unsupported Device" on page 6-4.
Determine if other required products have been installed.	"Other Products May be Required" on page 6-5.
Decide which esoteric device groups to define, and select a name for each group.	"Esoteric Device Groups and EDTs" on page 2-3. "What is an Esoteric Device Group?" on page 2-4. "Required Esoteric Device Groups" on page 7-5.
Decide how many eligible device tables to define and select an identifier for each table. **Note:** JES3 supports only the first EDT defined in an I/O configuration. Therefore, JES3 installations should define no more than one EDT per I/O configuration.	"Esoteric Device Groups and EDTs" on page 2-3. "JES3 Initialization Stream Checker Data" on page 1-11
Select the esoteric device groups and the generic device groups that you want to make eligible for VIO.	"What is a Virtual I/O (VIO) Eligible Device?" on page 2-6 Appendix A, "Generic Device Types" on page A-1
Decide what changes, if any, you will make to the device preference table for each EDT.	"What is the Device Preference Table?" on page 2-4. Appendix B, "IBM Provided Device Preference Table" on page B-1.
Select devices that you will allow NIP to use as consoles.	"What is the NIP Console Table?" on page 2-6. "Defining NIP Consoles" on page 7-9.
Select an identifier for the configuration.	"MVS Configuration Program Output" on page 1-9. "Assigning an I/O Configuration Identifier" on page 8-3.
Decide whether you want the MVS configuration program to create data that the JES3 initialization stream checker will use.	"JES3 Initialization Stream Checker Data" on page 1-11.
Decide which data sets to use for: • The input statements. • The I/O configuration that will be built. • The messages you will receive. • The reports you will receive. • The JES3 initialization stream data.	"Data Set Descriptions" on page 1-5.
Decide which MVS configuration program mode to use. The mode determines the type of output that you will receive.	"Selecting the Mode of Operation" on page 8-15.

Overview of the I/O Configuration Management Process

You should do the following activities only after you have completed your plan. These activities are fully describes in the procedures found in Chapter 4, "Managing I/O Configurations" on page 4-1. They are mentioned here just to give you a complete picture of the I/O configuration planning and definition process.

- **Code the input statements**
  These statements, which contain information developed during the planning step, form the I/O configuration definition. They are input to both the MVS configuration program and IOCP.
- **Modify the IBM-provided catalog procedure or the SAMPLIB JCL**
  If you are going to use the IBM-provided catalog procedure or SAMPLIB JCL, make any required changes or overrides.
- **Execute the MVS configuration program**
  Defines the I/O configuration to MVS.
- **Verify the I/O configuration**
  Verify the I/O configuration by comparing the MVS configuration program reports to your plan.
- **Execute the IOCP** - Defines the I/O configuration to the channel subsystem.

Chapter 4. Managing I/O Configurations

During the MVS/System Product install process, you must define at least one I/O configuration to MVS and to the channel subsystem. If you intend to use an I/O configuration that was previously defined through the SYSGEN process, you may do so by migrating the I/O configuration. Otherwise, you will have to define a new I/O configuration.

Migrating an I/O configuration means you use **IODEVICE** statements, **UNITNAME** statements and IOCP statements from the SYSGEN input stream. You merge these statements with other MVS configuration program statements, that you code, to create the MVS configuration program and IOCP input stream. You repeat this procedure for each I/O configuration you wish to migrate.

After the MVS/System Product is installed and its I/O configurations defined, you must maintain those I/O configurations. Maintaining I/O configurations means, defining new I/O configurations, updating existing I/O configurations or eligible device tables, or deleting I/O configurations or eligible device tables.

This chapter contains step-by-step procedures that explain how to:

- Migrate an I/O configuration previously defined through SYSGEN.
- Define a new I/O configuration.
- Update an existing I/O configuration.
- Delete an I/O configuration.
- Add, replace, or delete an eligible device table (EDT).
- Update the device preference table.
- Update the NIP console table.

Locating Information in this Chapter

This chapter contains the following major topics:

Using the Procedures

The first step in using any of the five procedures in this chapter is to determine which procedure to use. You can do this by using the decision table in "Selecting the Correct Procedure" on page 4-3.

After you have determined the procedure to use:

- Familiarize yourself with the procedure by reading it through completely. Also, look over the information referenced by each procedure step.

  Each procedure starts with an explanation of what the procedure does. This is followed by:

  - An explanation of when to do the procedure.
  - A list of the inputs that must be available before you use the procedure.
  - A description of the output that the procedure produces.

  Finally, there is a step-by-step explanation of each activity that you must do to complete the procedure. These explanations are brief. Therefore, for most activities, there are references to other topics or books where you can find more detailed information.

- Once you are familiar with the procedure, start at step 1 and follow the procedure step-by-step.
- There may be some steps that you will not need to do. In these cases, go to the next step.
- If you need to do a step but the procedure does not provide enough information, use the information referenced by the procedure step.

Note: If you are migrating from a release prior to MVS/SP2.2.0, before attempting to IPL MVS/SP3.1.0, you must run the MVS configuration program in IODEF mode to build an I/O configuration that MVS/SP3.1.0 can use. To build this configuration, use either of the following procedures:

- "Migrating an I/O Configuration" on page 4-4
- "Defining a New I/O Configuration" on page 4-17

If you are migrating from MVS/SP2.2.0, you can continue to use the I/O configurations that you previously used on MVS/SP2.2.0.

In either case, the I/O configuration must be available on the SYS1.NUCLEUS data during the IPL. Otherwise, the IPL will fail.

Selecting the Correct Procedure

The following decision table will help you decide which procedure to use. The left column identifies different tasks that you will do; the right column identifies the procedure that explains how to do that task.

Figure 4-1. Determining Which Procedure to Use

To Perform This Task:	Use This Procedure:
Migrate an I/O configuration Use this procedure if you have an I/O configuration, previously defined through SYSGEN, that you wish to use.	"Migrating an I/O Configuration" on page 4-4
Define a new I/O configuration Use this procedure to define a new I/O configuration.	"Defining a New I/O Configuration" on page 4-17
Update an I/O configuration: Use this procedure to update the following: • **I/O Device** - Add a new device, delete a device, or change device information. • **EDT** - Add a new EDT or update an existing EDT. • **Device preference table** - Change the device allocation order. • **Esoteric device group** - Add a new group, delete a group, or update a group. • **VIO-eligible devices** - Add a new group of devices, delete a group of devices, or update a group of devices. • **NIP console table** - Add devices to or remove devices from the table that identifies the devices that NIP may use as consoles.	"Updating an Existing I/O Configuration" on page 4-11
Delete an I/O configuration Use this procedure to delete an I/O configuration that you no longer need.	"Deleting an I/O Configuration" on page 4-23
Delete an eligible device table Use this procedure to delete an EDT that you no longer need.	"Deleting an Eligible Device Table (EDT)" on page 4-25

Migrating an I/O Configuration

This procedure migrates an MVS I/O configuration that was previously defined through the SYSGEN process. You must repeat this procedure for each such I/O configuration you wish to migrate.

> **Note:** If you are unfamiliar with this procedure, read it through completely before using it. You should also review the detailed information referred to by each step.

When to Do Procedure

- Anytime you are migrating an I/O configuration from a system that uses the SYSGEN process to a system that uses the MVS configuration program. As an alternative to migrating an I/O configuration, you can define a new I/O configuration as explained in, "Defining a New I/O Configuration" on page 4-17.
  - The *earliest* you can do this procedure is after you have the input listed below.
  - The Program Directory indicates the *latest* time in the install process that you can do this procedure.

Procedure Input

- The Program Directory for the MVS/System Product release that you are installing.
- The SYSGEN input stream that previously defined the configuration.
- The MVS configuration program, the IBM-provided cataloged procedure or the IBM-provided SAMPLIB member, and macro library (SYS1.MACLIB), must be available on the system where you intend to execute the MVS configuration program.
- If you must execute IOCP (See step 23 of this procedure):
  - IOCP and its input libraries.
  - The publication *IOCP User's Guide and Reference*.

Procedure Output

- The MVS configuration program writes the internal representation of the I/O configuration to IEANCTxx, IOSIITxx, IOSUCBxx, and IEFEDTxx.
  - The MVS configuration program writes one each of the following members: IEANCTxx, IOSIITxx, and IOSUCBxx.
  - The MVS configuration program writes one IEFEDTxx member for *each* eligible device table.
  - The MVS configuration program writes all of these members to the data set defined by the **SYSLMOD** DD statement.
- If you execute IOCP, IOCP writes the channel subsystem's internal representation of the I/O configuration to the IOCDS.
- Reports and messages.

Procedure Activities

Step	Migrating an I/O Configuration	Reference
(1)	**Plan the migration**: Review the current I/O configuration and define any desired changes. Document the changes in your configuration plan.	Chapter 3, "Planning an I/O Configuration" on page 3-1
(2)	**Copy the existing SYSGEN input stream**: Copy the input stream into a data set that you can edit. Use the program normally used to copy files at your installation. This step saves the SYSGEN input stream in case you need to refer to it later.	None
(3)	**Edit the copy of the SYSGEN input stream**: (Use the editor normally used at your installation) • Save statements — **ID**, **CHPID**, **CNTLUNIT**, **IODEVICE** and **UNITNAME**. • If you plan to use information from the DEVPREF parameter of the **SCHEDULR** statement, save the **SCHEDULR** statement. • If desired, save comment statements. • Delete all other statements including the **END** statement. • Review your plan and delete any of the saved statements that you no longer need. The statements that remain form a base on which to develop the MVS configuration program input stream.	"Using SYSGEN Input Statements to Migrate a Configuration" on page 8-2

Step	Migrating an I/O Configuration	Reference
(4)	**Review recommended order of the MVS configuration program input stream**: • IOCONFIG • ID • CHPID • CNTLUNIT • IODEVICE • EDT • UNITNAME • NIPCON **Notes:** 1. EDT statements, if used, *must* immediately precede their related UNITNAME statements. 2. Assembler comment statements may appear anywhere in the input stream. 3. *Do not* include an END statement or any other statement not listed above in the input stream.	"Ordering Statements in the Input Stream" on page 8-14 Appendix D, "MVS Configuration Program Example" on page D-1
(5)	**Review input stream example**: This sample MVS configuration program input stream will help you become familiar with statement formats and input stream organization.	Appendix D, "MVS Configuration Program Example" on page D-1
(6)	**Review the MVS configuration program input statements**: This review will help you understand the purpose of each statement and will allow you to become familiar with statement parameters and the rules for coding statements.	Chapter 5, "MVS Configuration Program Input Statement Reference" on page 5-1
(7)	**Assign the I/O configuration identifier**: • Use the ID parameter on the IOCONFIG statement. • Include the statement in the input stream.	"Assigning an I/O Configuration Identifier" on page 8-3 "IOCONFIG Statement" on page 5-7
(8)	**Define the IOCP definitions**: • Make planned IOCP changes to the **ID**, **CHPID**, **CNTLUNIT**, and **IODEVICE** statements previously saved. • If your plan calls for additional IOCP definitions, include the required statements in the input stream.	*IOCP User's Guide and Reference*

Step	Migrating an I/O Configuration	Reference
(9)	**Define the I/O devices:** • Make planned changes to the IODEVICE statements saved in step 3. • If the IODEVICE statements you saved contain any of the following parameters (AP, DEVTYPE, ERRTAB, EXPBFR, OBRCNT, or OPTCHAN), it is suggested you remove those parameters. If you allow them to remain, the MVS configuration program ignores the parameters, issues informational messages but, otherwise, works normally. • If any saved IODEVICE statement specifies UNIT=DUMMY, see "Migrating a Dummy Device" on page 8-2. • To define additional devices, include the appropriate IODEVICE statements in the input stream.	Chapter 6, "Defining I/O Devices" on page 6-1 "Channel-to-Channel Adapter Definitions" on page 6-7 "DASD Definition" on page 6-9 "Display Device Definitions" on page 6-25 "Magnetic Tape Device Definitions" on page 6-47 "Miscellaneous I/O Device Definitions" on page 6-54 "MSS Device Definitions" on page 6-64 "Telecommunication Device Definitions" on page 6-67 "Unit Record Device Definitions" on page 6-92 "Migrating a Dummy Device" on page 8-2 "IODEVICE Statement" on page 5-9
(10)	**Define eligible device tables (EDTs):** • To define one EDT with no changes to the device preference table: You can code an EDT statement or omit it. If you omit the EDT statement, the EDT identifier defaults to the identifier (ID=) on the IOCONFIG statement. • To define one EDT with changes to the device preference table: Code one EDT statement. If you omit the ID parameter from the EDT statement, the EDT identifier defaults to the identifier on the IOCONFIG statement. • To define multiple EDTs: Code one EDT statement for each EDT. You may omit the ID parameter from one (but no more) of the EDT statements. If you omit the ID parameter, the EDT identifier defaults to the identifier on the IOCONFIG statement. **Notes:** 1. No two EDTs can have the same identifier. If you specify the identifier of an existing EDT, the MVS configuration program replaces the existing EDT with the new EDT. 2. Ensure that UNITNAME statements immediately follow their corresponding EDT statement. 3. JES3 supports only one EDT per I/O configuration. Therefore, JES3 installations should define only one EDT per I/O configuration.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 "EDT Statement" on page 5-4

Step	Migrating an I/O Configuration	Reference
(11)	**Define device preference table updates**: • Code any updates on the DEVPREF parameter of the corresponding EDT statement. • If you saved the SCHEDULR statement in step 3, delete it. **Note:** In the SYSGEN input stream, changes to the device preference table were coded on the DEVPREF parameter of the SCHEDULR statement.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 Appendix B, "IBM Provided Device Preference Table" on page B-1 Appendix A, "Generic Device Types" on page A-1 "EDT Statement" on page 5-4
(12)	**Define esoteric device groups**: • Make any planned changes to the UNITNAME statements saved in step 3. • To define additional esoteric device groups, code the required UNITNAME statements. Indicate, on the UNITNAME statements, whether the group is to be eligible for VIO. There will be one set of UNITNAME statements for each EDT. • Place each set of UNITNAME statements immediately after the EDT statement to which they correspond. If the input stream contains no EDT statements, place the UNITNAME statements immediately after the last IODEVICE statement. **Note:** To determine which esoteric device groups your installation requires, review the cataloged procedures used at your installation (check the UNIT parameter on DD statements). If you use the IBM-provided SYS1.PROCLIB, you may have to define esoteric device groups SYSDA and SYSSQ.	"Defining Esoteric Device Groups" on page 7-5 "Required Esoteric Device Groups" on page 7-5 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17
(13)	**Identify VIO-eligible generic device types**: • Identify, on a UNITNAME statement, each generic device type that you want MVS to consider for VIO allocations. • Group the UNITNAME statements after the EDT statement to which they correspond.	"Identifying VIO-Eligible Generic Device Types" on page 7-7 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17
(14)	**Define NIP consoles**: Code device numbers on one or more NIPCON statements. • NIP requires at least one console. • If you define multiple consoles, NIP attempts to use them in the defined order. • IBM recommends that you define the master console as the first NIP console.	"Defining NIP Consoles" on page 7-9 "NIPCON Statement" on page 5-15

Step	Migrating an I/O Configuration	Reference
(15)	**Define cataloged procedure overrides or SAMPLIB JCL changes:** • If you use the cataloged procedure, include desired overrides in the invocation JCL. • If you use the SAMPLIB JCL, change it as needed.	"Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 "Invoking the MVS Configuration Program" on page 8-17 *JCL Reference* and *JCL Users Guide*
(16)	**Specify SYSIN and JES3 data set names:** • SYSIN data set: – Required data set. – If you use the cataloged procedure, specify the data set name on the PROC statement's DEFDATA parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSIN DD statement. – The SYSIN DD statement defines the data set. • JES3 data set: – Required only if you want the MVS configuration program to write data that the JES3 initialization stream checker can read. – If you use the cataloged procedure, specify the data set name on the PROC statement's JES3 parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSJES3 DD statement. – The SYSJES3 DD statement defines the data set.	"Identifying the SYSIN and JES3 Data Sets" on page 8-13 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 *JES3 Initialization and Tuning*.
(17)	**Select the mode of operation:** • To build the migrated configuration, select IODEF mode. • To perform a syntax check and semantics check on the input statements, and to receive the reports without building the configuration, select VALIDATE mode. • If you use the catalog procedure, specify mode on the TYPE parameter (the default is VALIDATE). • If you use the SAMPLIB JCL, specify mode on the PARM parameter of the EXEC statement (the default is VALIDATE).	"Selecting the Mode of Operation" on page 8-15 "Invoking the MVS Configuration Program" on page 8-17 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4
(18)	**Invoke the MVS configuration program**	"Invoking the MVS Configuration Program" on page 8-17
(19)	**Determine if the run was successful:** • Check the JES job log and the files SYSASMP, SYSLKEDP, SYSLOUT, and SYSPRINT for messages that indicate possible errors. • Check the SYSRPT file to ensure that you received the expected reports. • If there are no error messages and you received the expected reports, continue with the next numbered step in this procedure. • If there are error messages or reports are missing, follow the instructions given in the topic "Verifying MVS Configuration Program Output" on page 8-20.	Chapter 10, "Debugging, Messages, and Codes" on page 10-1 "Verifying MVS Configuration Program Output" on page 8-20 Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1

Step	Migrating an I/O Configuration	Reference
(20)	**Verify the configuration:** To confirm that you built the I/O configuration that you intended to build, compare your configuration plan to the MVS configuration program reports.	Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1.
(21)	**If you selected VALIDATE mode:** • If you now wish to build the configuration, repeat the procedure from step 17, this time selecting IODEF mode.	None
(22)	**Copy the I/O configuration members to the SYS1.NUCLEUS data set:** • Do this step *only* if the I/O configuration output data set (defined by the //SYSLMOD DD statement) is a data set other than the SYS1.NUCLEUS data set of the system on which the configuration will be used. • Use the utility program IEBCOPY or a similar program.	*Utilities*
(23)	**Invoke the IOCP:** You must invoke the IOCP if: • You do not have an IOCDS on which the I/O configuration is defined. • You have an IOCDS but you have added, deleted, or updated one or more of the following statements: CHPID, CNTLUNIT, ID, or IODEVICE.	See the publication *IOCP User's Guide and Reference*

Note: If you wish to validate an EDT, see Appendix E, "Eligible Device Table Verification Service" on page E-1.

Updating an Existing I/O Configuration

This procedure updates an existing I/O configuration that was previously migrated or defined through the MVS configuration program.

> **Note:** If you are unfamiliar with this procedure, read it through completely before using it. You should also review the detailed information referred to by each step.

When to Do Procedure

Anytime there is a need to update an I/O configuration previously migrated or defined through the MVS configuration program.

Procedure Input

- The combined MVS configuration program/IOCP input stream that currently defines the configuration.
- The data set that contains the MVS internal representation of the I/O configuration.
- The MVS configuration program, the IBM-provided cataloged procedure or the IBM-provided SAMPLIB member, and macro library (SYS1.MACLIB), must be available on the system where you intend to execute the MVS configuration program.
- If you must execute IOCP (See step 21 of this procedure):
  - IOCP and its input libraries.
  - The publication *IOCP User's Guide and Reference*.

Procedure Output

- The MVS internal representation of the I/O configuration:.
  - If you execute the MVS configuration program in EDT mode, the MVS configuration program writes only IEFEDTxx members. (it writes a separate member for each EDT.)
  - If you execute the MVS configuration program in IODEF mode, the MVS configuration program writes a separate IEFEDTxx member for each EDT and one each of the members IEANCTxx, IOSIITxx, and IOSUCBxx.
  - The MVS configuration program writes these members to the data set defined by the SYSLMOD DD statement.
- If you execute IOCP, IOCP writes the channel subsystem's internal representation of the I/O configuration data to the IOCDS.
- Reports and messages.

Procedure Activities

Step	Updating an I/O Configuration	Reference
(1)	**Plan the update:** • Review the existing configuration plan and configuration reports. • Define the changes and document them in your configuration plan.	Chapter 3, "Planning an I/O Configuration" on page 3-1 "Requesting Reports and Validating the Input Stream" on page 9-2
(2)	**Review recommended order of MVS configuration program input stream:** • IOCONFIG • ID • CHPID • CNTLUNIT • IODEVICE • EDT • UNITNAME • NIPCON **Notes:** 1. EDT statements, if used, *must* immediately precede their related UNITNAME statements. 2. Assembler comment statements may appear anywhere in the input stream. 3. *Do not* include an END statement or any other statement not listed above in the input stream.	"Ordering Statements in the Input Stream" on page 8-14
(3)	**Review input stream example:** This sample MVS configuration program input stream will help you become familiar with statement formats and input stream organization.	Appendix D, "MVS Configuration Program Example" on page D-1
(4)	**Review the MVS configuration program input statements:** This review will help you understand the purpose of each statement and will allow you to become familiar with statement parameters and the rules for coding statements.	Chapter 5, "MVS Configuration Program Input Statement Reference" on page 5-1

Step	Updating an I/O Configuration	Reference
(5)	**Update I/O device definitions**: • To add a device — include a new IODEVICE statement. • To delete a device — remove the corresponding IODEVICE statement. **Note:** Also remove the device number from all UNITNAME statements and NIPCON statements on which it appears. Be especially careful to remove device numbers that are specified on the IODEVICE statement through the use of the *number-of-units* subparameter, they are easy to miss. When you finish this step, be sure the input stream contains at least one NIPCON statement. • To change device information — change the information on the corresponding IODEVICE statement.	Chapter 6, "Defining I/O Devices" on page 6-1 "Channel-to-Channel Adapter Definitions" on page 6-7 "DASD Definition" on page 6-9 "Display Device Definitions" on page 6-25 "Magnetic Tape Device Definitions" on page 6-47 "Miscellaneous I/O Device Definitions" on page 6-54 "MSS Device Definitions" on page 6-64 "Telecommunication Device Definitions" on page 6-67 "Unit Record Device Definitions" on page 6-92 "IODEVICE Statement" on page 5-9
(6)	**Update an esoteric device group**: • To add a device to a group — include the device number(s) on the corresponding UNITNAME statement. • To delete a device from a group — remove the device number(s) from the corresponding UNITNAME statement.	"Defining Esoteric Device Groups" on page 7-5 "UNITNAME Statement" on page 5-17
(7)	**Delete an esoteric device group**: • Remove the corresponding UNITNAME statement(s) from the input stream. **Note:** To ensure that you do not delete an esoteric device group that you still require, review the cataloged procedures used at your installation (check the UNIT parameter on DD statements). If your installation uses the IBM-provided SYS1.PROCLIB, you probably require esoteric device groups SYSDA and SYSSQ.	None

Step	Updating an I/O Configuration	Reference
(8)	**Update a device preference table definition**: • To define updates, use the DEVPREF parameter on the appropriate EDT statement. • If the input stream contains no EDT statement, you must code one.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 Appendix B, "IBM Provided Device Preference Table" on page B-1 Appendix A, "Generic Device Types" on page A-1 "EDT Statement" on page 5-4
(9)	**Define an additional eligible device table (EDT)**: • If the input stream contains no EDT statements: – Code two EDT statements: One for the previously defined EDT and one for the additional EDT. – Assign each EDT an identifier. – To change the device preference table definition, use the DEVPREF parameter on the EDT statement. • If the input stream contains EDT statements: – Code an EDT statement for the additional EDT. – Assign the EDT an identifier. – To change the device preference table definition, use the DEVPREF parameter on the EDT statement. **Notes:** 1. No two EDTs can have the same identifier. If you specify the identifier of an existing EDT, the MVS configuration program replaces the existing EDT with the new EDT. 2. Ensure that UNITNAME statements immediately follow their corresponding EDT statement. 3. JES3 supports only one EDT per I/O configuration. Therefore, JES3 installations should define only one EDT per I/O configuration.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 "EDT Statement" on page 5-4
(10)	**Define a new esoteric device group**: • Define the group on a UNITNAME statement. Indicate whether the group is to be eligible for VIO. • Place the UNITNAME statement into the input stream following the corresponding EDT statement.	"Defining Esoteric Device Groups" on page 7-5 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17
(11)	**Identify VIO-eligible generic device types**: • Identify, on a UNITNAME statement, each generic device type that you want MVS to consider for VIO allocations. • Group the UNITNAME statements after the EDT statement to which they correspond.	"Identifying VIO-Eligible Generic Device Types" on page 7-7 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17

Step	Updating an I/O Configuration	Reference
(12)	**Update NIP console definition**: Add or remove device numbers on the NIPCON statements. • NIP requires at least one console. • If you define multiple consoles, NIP attempts to use them in the defined order. • IBM recommends that you define the master console as the first NIP console.	"Defining NIP Consoles" on page 7-9. "NIPCON Statement" on page 5-15
(13)	**Define cataloged procedure overrides or SAMPLIB JCL changes**: • If you use the cataloged procedure, include desired overrides in the invocation JCL. • If you use the SAMPLIB JCL, change it as needed.	"Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 "Invoking the MVS Configuration Program" on page 8-17 *JCL Reference* and *JCL Users Guide*
(14)	**Specify SYSIN and JES3 data set names**: • SYSIN data set: – Required data set. – If you use the cataloged procedure, specify the data set name on the PROC statement's DEFDATA parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSIN DD statement. – The SYSIN DD statement defines the data set. • JES3 data set: – Required only if you want the MVS configuration program to write data that the JES3 initialization stream checker can read. – If you use the cataloged procedure, specify the data set name on the PROC statement's JES3 parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSJES3 DD statement. – The SYSJES3 DD statement defines the data set.	"Identifying the SYSIN and JES3 Data Sets" on page 8-13 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 *JES3 Initialization and Tuning*.
(15)	**Select the mode of operation**: • If you added, deleted, or changed UNITNAME statements, EDT statements, or both, and made no other changes, you can select either EDT mode or IODEF mode. • For all other changes, select IODEF mode. • To perform a syntax check and semantics check on the input statements, and to receive the reports without building the configuration, select VALIDATE mode. • If you use the cataloged procedure, specify mode on the TYPE parameter (the default is VALIDATE). • If you use the SAMPLIB JCL, specify mode on the PARM parameter of the EXEC statement (the default is VALIDATE).	"Selecting the Mode of Operation" on page 8-15 "Invoking the MVS Configuration Program" on page 8-17 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4

Step	Updating an I/O Configuration	Reference
(16)	**Invoke the MVS configuration program**	"Invoking the MVS Configuration Program" on page 8-17
(17)	**Determine if the run was successful:** • Check the JES job log and the files SYSASMP, SYSLKEDP, SYSLOUT, and SYSPRINT for messages that indicate errors. • Check the SYSRPT file to ensure that you received the expected reports. • If there are no error messages and you received the expected reports, continue with the next numbered step in this procedure. • If there are error messages or reports are missing, follow the instructions given in the topic "Verifying MVS Configuration Program Output" on page 8-20.	Chapter 10, "Debugging, Messages, and Codes" on page 10-1 "Verifying MVS Configuration Program Output" on page 8-20 Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1
(18)	**Verify the configuration:** To confirm that you built the I/O configuration that you intended to build, compare your configuration plan to the MVS configuration program reports.	Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1.
(19)	**If you selected VALIDATE mode:** • If you now wish to build the configuration, repeat the procedure from step 15, this time selecting IODEF mode or EDT mode as appropriate.	None
(20)	**Copy the I/O configuration members to the SYS1.NUCLEUS data set:** • Do this step *only* if the I/O configuration output data set (defined by the //SYSLMOD DD statement) is a data set other than the SYS1.NUCLEUS data set of the system on which the configuration will be used. • Use the utility program IEBCOPY or a similar program.	*Utilities*
(21)	**Invoke the IOCP**: You must invoke the IOCP if: • You added, deleted, or changed an IOCP statement or an IODEVICE statement.	See the publication *IOCP User's Guide and Reference*

Note: If you wish to validate an EDT, see Appendix E, "Eligible Device Table Verification Service" on page E-1.

Defining a New I/O Configuration

This procedure defines a new I/O configuration. A new I/O configuration is one whose identifier is currently undefined.

> **Note:** If you are unfamiliar with this procedure, read it through completely before using it. You should also review the detailed information referred to by each step.

When to Do Procedure

Anytime you wish to define a new I/O configuration.

Procedure Input

- The MVS configuration program, the IBM-provided cataloged procedure or the IBM-provided SAMPLIB member, and macro library (SYS1.MACLIB), must be available on the system where you intend to execute the MVS configuration program.
- IOCP and its input libraries.
- The publication *IOCP User's Guide and Reference.*

Procedure Output

- The MVS configuration program writes the internal representation of the I/O configuration to IEANCTxx, IOSIITxx, IOSUCBxx, and IEFEDTxx.
  - The MVS configuration program writes one each of the following members: IEANCTxx, IOSIITxx, and IOSUCBxx.
  - The MVS configuration program writes one IEFEDTxx member for *each* EDT.
  - The MVS configuration program writes all of these members to the data set defined by the **SYSLMOD** DD statement.
- IOCP writes the channel subsystem's internal representation of the I/O configuration to the IOCDS.
- Reports and messages.

Procedure Activities

Step	Defining a New I/O Configuration	Reference
(1)	**Plan the configuration:** Document the I/O configuration in your configuration plan.	Chapter 3, "Planning an I/O Configuration" on page 3-1
(2)	**Review recommended order of MVS configuration program input stream:** • IOCONFIG • ID • CHPID • CNTLUNIT • IODEVICE • EDT • UNITNAME • NIPCON **Notes:** 1. EDT statements, if used, *must* immediately precede their related UNITNAME statements. 2. Assembler comment statements may appear anywhere in the input stream. 3. *Do not* include an END statement or any other statement not listed above in the input stream.	"Ordering Statements in the Input Stream" on page 8-14 Appendix D, "MVS Configuration Program Example" on page D-1
(3)	**Review input stream example:** This sample input stream will help you become familiar with statement formats and input stream organization.	Appendix D, "MVS Configuration Program Example" on page D-1
(4)	**Review the MVS configuration program input statements:** This review will help you understand the purpose of each statement and will allow you to become familiar with statement parameters and the rules for coding statements.	Chapter 5, "MVS Configuration Program Input Statement Reference" on page 5-1
(5)	**Assign the I/O configuration identifier:** • Use the ID parameter on the IOCONFIG statement. • Include the statement in the input stream.	"Assigning an I/O Configuration Identifier" on page 8-3 "IOCONFIG Statement" on page 5-7
(6)	**Define the IOCP definitions:** Include the IOCP statements ID, CHPID, and CNTLUNIT in the input stream.	*IOCP User's Guide and Reference*

Step	Defining a New I/O Configuration	Reference
(7)	**Define I/O devices**: • Define each device on an IODEVICE statement. • Include the IODEVICE statements in the input stream.	Chapter 6, "Defining I/O Devices" on page 6-1 "Channel-to-Channel Adapter Definitions" on page 6-7 "DASD Definition" on page 6-9 "Display Device Definitions" on page 6-25 "Magnetic Tape Device Definitions" on page 6-47 "Miscellaneous I/O Device Definitions" on page 6-54 "MSS Device Definitions" on page 6-64 "Telecommunication Device Definitions" on page 6-67 "Unit Record Device Definitions" on page 6-92 "IODEVICE Statement" on page 5-9
(8)	**Define eligible device tables (EDTs)**: If your plan calls for: • One EDT with no changes to the device preference table: You may omit the EDT statement. If you omit the EDT statement, the EDT identifier defaults to the identifier (ID=) on the IOCONFIG statement. • One EDT with changes to the device preference table: Code one EDT statement. If you omit the ID parameter from the EDT statement, the EDT identifier defaults to the identifier on the IOCONFIG statement. • Multiple EDTs: Code one EDT statement for each EDT. You may omit the ID parameter from one (but no more) of the EDT statements. If you omit the ID parameter, the EDT identifier defaults to the identifier on the IOCONFIG statement. **Notes:** 1. No two EDTs can have the same identifier. If you specify the identifier of an existing EDT, the MVS configuration program replaces the existing EDT with the new EDT. 2. Ensure that UNITNAME statements immediately follow their corresponding EDT statement. 3. JES3 supports only one EDT per I/O configuration. Therefore, JES3 installations should define only one EDT per I/O configuration.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 "EDT Statement" on page 5-4

Step	Defining a New I/O Configuration	Reference
(9)	**Define device preference table updates**: To define updates, use the DEVPREF parameter on the EDT statements to which the updates correspond.	Chapter 7, "Defining Eligible Device Tables and NIP Consoles" on page 7-1 Appendix B, "IBM Provided Device Preference Table" on page B-1 Appendix A, "Generic Device Types" on page A-1 "EDT Statement" on page 5-4
(10)	**Define esoteric device groups**: • Define each group on a UNITNAME statement. Indicate whether the group is to be eligible for VIO. • Place the UNITNAME statements after the EDT statement to which they correspond. • If the input stream contains no EDT statements, place the UNITNAME statements after the last IODEVICE statement. **Note:** To determine which esoteric device groups your installation requires, review the cataloged procedures used at your installation (check the UNIT parameter on DD statements). If you use the IBM-provided SYS1.PROCLIB, you may have to define esoteric device groups SYSDA and SYSSQ.	"Defining Esoteric Device Groups" on page 7-5 "Required Esoteric Device Groups" on page 7-5 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17
(11)	**Identify VIO-eligible generic device types**: • Identify, on a UNITNAME statement, each generic device type that you want MVS to consider for VIO allocations. • Place the UNITNAME statements after the EDT statement to which they correspond.	"Identifying VIO-Eligible Generic Device Types" on page 7-7 "Grouping EDT and UNITNAME Statements" on page 7-8 "UNITNAME Statement" on page 5-17
(12)	**Define NIP consoles**: Code device numbers on NIPCON statements. • NIP requires at least one console. • If you define multiple consoles, NIP attempts to use them in the defined order. • IBM recommends that you define the master console as the first NIP console.	"Defining NIP Consoles" on page 7-9. "NIPCON Statement" on page 5-15
(13)	**Define cataloged procedure overrides or SAMPLIB JCL changes**: • If you use the cataloged procedure, include desired overrides in the invocation JCL. • If you use the SAMPLIB JCL, change it as needed.	"Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 "Invoking the MVS Configuration Program" on page 8-17 *JCL Reference* and *JCL Users Guide*

Step	Defining a New I/O Configuration	Reference
(14)	**Specify SYSIN and JES3 data set names:** • SYSIN data set: – Required data set. – If you use the cataloged procedure, specify the data set name on the PROC statement's DEFDATA parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSIN DD statement. – The SYSIN DD statement defines the data set. • JES3 data set: – Required only if you want the MVS configuration program to write data that the JES3 initialization stream checker can read. – If you use the cataloged procedure, specify the data set name on the PROC statement's JES3 parameter. – If you use the SAMPLIB JCL, specify the data set name on the DSN parameter of the SYSJES3 DD statement. – The SYSJES3 DD statement defines the data set.	"Identifying the SYSIN and JES3 Data Sets" on page 8-13 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4 *JES3 Initialization and Tuning*.
(15)	**Select the mode of operation:** • To build the new configuration, select IODEF mode. • To perform a syntax check and semantics check on the input statements, and to receive the reports without building the configuration, select VALIDATE mode. • If you use the cataloged procedure, specify mode on the TYPE parameter (the default is VALIDATE). • If you use the SAMPLIB JCL, specify mode on the PARM parameter of the EXEC statement (the default is VALIDATE).	"Selecting the Mode of Operation" on page 8-15 "Invoking the MVS Configuration Program" on page 8-17 "SYS1.PROCLIB(MVSCP) Cataloged Procedure" on page 8-5 "Using the Cataloged Procedure or the SAMPLIB JCL" on page 8-4
(16)	**Invoke the MVS configuration program**	"Invoking the MVS Configuration Program" on page 8-17
(17)	**Determine if the run was successful:** • Check the JES job log and the files SYSASMP, SYSLKEDP, SYSLOUT, and SYSPRINT for messages that indicate possible errors. • Check the SYSRPT file to ensure that you received the expected reports. • If there are no error messages and you received the expected reports, continue with the next numbered step in this procedure. • If there are error messages or reports are missing, follow the instructions given in the topic "Verifying MVS Configuration Program Output" on page 8-20.	Chapter 10, "Debugging, Messages, and Codes" on page 10-1 "Verifying MVS Configuration Program Output" on page 8-20 Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1
(18)	**Verify the configuration:** To confirm that you built the I/O configuration that you intended to build, compare your configuration plan to the MVS configuration program reports.	Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1.

Step	Defining a New I/O Configuration	Reference
(19)	**If you selected VALIDATE mode:** • If you now wish to build the configuration, repeat the procedure from step 15, this time selecting IODEF mode.	None
(20)	**Copy the I/O configuration members to the SYS1.NUCLEUS data set:** • Do this step *only* if the I/O configuration output data set (defined by the //SYSLMOD DD statement) is a data set other than the SYS1.NUCLEUS data set of the system on which the configuration will be used. • Use the utility program IEBCOPY or a similar program.	*Utilities*
(21)	**Invoke the IOCP**	See the publication *IOCP User's Guide and Reference*

Note: If you wish to validate an EDT, see Appendix E, "Eligible Device Table Verification Service" on page E-1.

Deleting an I/O Configuration

This procedure deletes an I/O configuration definition and the corresponding I/O configuration data.

> **Note:** If you are unfamiliar with this procedure, read it through completely before using it. You should also review the detailed information referred to by each step.

When to Do Procedure

Anytime there is an I/O configuration that is no longer needed.

> **Warning:**
> Do not delete a system's last I/O configuration.
> If you do, you will be unable to re-IPL MVS on that system until you define another configuration.

Procedure Input

- The SYS1.NUCLEUS data set that contains the I/O configuration data.
- The data set that contains the MVS configuration program input statements (the I/O configuration definition).
- The publication *Utilities*.

Procedure Output

- The I/O configuration definition is scratched from its data set.
- The I/O configuration data (members IEANCTxx, IOSIITxx, IOSUCBxx, and IEFEDTxx) is scratched from the SYS1.NUCLEUS data set.

Procedure Activities

Step	Delete an I/O Configuration	Reference
(1)	**Scratch members IEANCTxx, IOSIITxx, and IOSUCBxx**: • The value xx is the identifier of the I/O configuration. It is found on the IOCONFIG statement. • Use the utility program IEHPROGM or a similar program.	*Utilities*
(2)	**Scratch IEFEDTxx member(s)**: • There is one member for each EDT defined. Scratch each member. • The value xx corresponds to the identifier (ID) specified on each EDT statement. • If there are no EDT statements, or there is an EDT statement that omits the ID parameter, the value xx corresponds to the identifier specified on the IOCONFIG statement. • Use utility program IEHPROGM or a similar program.	*Utilities*
(3)	**Scratch the I/O configuration definition**: • Use utility program IEHPROGM or a similar program.	*Utilities*

Deleting an Eligible Device Table (EDT)

This procedure deletes an EDT member (IEFEDTxx) and the corresponding statements from the I/O configuration definition.

> **Note:** If you are unfamiliar with this procedure, read it through completely before using it. You should also review the detailed information referred to by each step.

When to Do Procedure

Anytime the installation has an EDT that it no longer needs.

Procedure Input

- The SYS1.NUCLEUS data set that contains the I/O configuration data.
- The data set that contains the MVS configuration program input statements (the I/O configuration definition).
- The publication *Utilities*.

Procedure Output

- The EDT is deleted from the I/O configuration definition.
- The I/O configuration data (member IEFEDTxx is scratched from the SYS1.NUCLEUS data set.

Procedure Activities

Step	Delete an EDT	Reference
(1)	**Scratch the IEFEDTxx member:** • The value xx is the identifier of the EDT. • Use the utility program IEHPROGM or a similar program. **Note:** *xx* is the identifier assigned to the EDT at the time the EDT was built.	*Utilities*
(2)	**Update the I/O configuration data:** • Remove, from the I/O configuration data, the EDT statement and UNITNAME statements that correspond to the IEFEDTxx member scratched. • Use the edit program normally used at your installation for this purpose.	None
(3)	**Print a Configuration Report:** • The report reflects the configuration with the EDT deleted. • For future reference, file the report with the configuration plan.	"Requesting Reports and Validating the Input Stream" on page 9-2

Chapter 5. MVS Configuration Program Input Statement Reference

There are five types of input statements that you use to define an I/O configuration to the MVS configuration program. The statements are:

- EDT
- IOCONFIG
- IODEVICE
- NIPCON
- UNITNAME

This chapter:

- Explains the conventions and syntax that this book uses to describe the statements.
- Contains a description of each statement and its parameters.

Locating Information in this Chapter

This chapter contains the following major topics:

Coding Rules for Input Statements

The rules for coding input statements are the same rules that must be followed to code Assembler statements. The following paragraphs are a summary of these rules as stated in *Assembler H Version 2 Application Programmer's Language Reference* and *Assembler H Version 2 Application Programming Guide*.

- Code each statement in columns 1 through 71 only.
- Use one or more blanks to separate each field, *name*, **Operation**, **Operand**, and *comments* from adjacent fields.
- Use column 72 only to indicate continuation.
- Use columns 73 through 80 for sequence numbers or other installation-defined information. The MVS configuration program does not use this information.

Statement Format

Each input statement must conform to the following format:

Name	Operation	Operand	Comments
name	Statement name	Required and optional parameters	Optional

name

Symbolically identifies the statement. The MVS configuration program does not use this optional field. If included, *name* must contain from 1 through 8 alphanumeric or national characters, the first of which must be an alphabetic or national character.

National characters are represented by the hexadecimal codes X'5B', X'7B', and X'7C'. In U.S. English EBCDIC code, these hexadecimal codes represent the characters $, #, and @, respectively. In other national languages, these codes represent different characters.

Do not code X'7C' (@) as the first character in *name*, because the MVS configuration program uses it internally.

If you code *name*, it must start in column one.

Operation

Identifies the specific input statement. For example, IODEVICE, or UNITNAME.

This field must not start in column one. If you also code the Name field, separate the Name and Operation fields by at least one blank.

Operand

Contains statement parameters separated by commas.

A parameter consists of a keyword followed by an equal (=) sign and the keyword value. The keyword value may be a single value or a list of values. If you code a list of values, use commas to separate the values and enclose the entire list in parentheses.

This field ends with one or more blanks placed after the last parameter.

comment

Comments are optional. Use one or more blanks to separate the last operand from the comment. To place a comment only on a statement, code an asterisk in the first column of the statement and follow the asterisk with the comment.

Continuing a Statement

To continue an input statement:

- Interrupt the statement that is to be continued (the continued statement) either in column 71 or after any comma.
- Code a non-blank character in column 72.
- Start coding the next statement (the continuation statement) in column 16.
- You can code comments on a continued statement or on a continuation statement.

The Syntax Used to Describe Input Statements

This book uses the following notation to describe the format of the input statements. This notation is not part of the statement language. It is merely a tool that serves to describe the statement format and structure.

- Brackets [ ] indicate an optional parameter.
- Braces { } indicate a choice of entry; unless a default is indicated, you must select one of the entries.
- Items separated by a vertical bar | represent alternative items. Select no more than one of these items.
- An ellipsis (...) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- Other punctuation (parentheses, commas, spaces, etc.) must be entered as shown. A space is indicated by a blank.
- **BOLDFACE** type indicates the exact characters to be entered. Such items must be entered exactly as illustrated.
- *Lowercase italicized* type identifies fields that you must supply.
- ***BOLDFACE UNDERSCORED*** type indicates a default value. If you omit the parameter, the MVS configuration program uses the default value.
- If you code more than one value for a parameter, enclose the values in parentheses **( )**. If you code only one value, omit the parentheses.

EDT Statement

Use the **EDT** statement to:

- Assign an identifier to an eligible device table (EDT)
- Redefine the order that MVS is to use when attempting to satisfy an allocation request

To MVS, the identifier distinguishes one eligible device table from another. The system operator may use the identifier to select an eligible device table during IPL processing.

The order of devices in the IBM-supplied device preference table is shown in Appendix B, "IBM Provided Device Preference Table" on page B-1.

For a list of valid generic names, see Appendix A, "Generic Device Types" on page A-1.

Rules for coding input statements are in "Coding Rules for Input Statements" on page 5-2.

[name]	**EDT**	[**ID=** *xx*] [,**DEVPREF**=(genericname[,genericname]...)]

name
A 1-to-8 character name. The MVS configuration program ignores this name. You may wish to code it, however, to help you identify your input statements.

ID = *xx*
Assigns the identifier, *xx*, to an eligible device table. The identifier *xx* can be any two-digit alphanumeric value. You can assign each digit a value 0-9 or A-Z.

Default:

- The identifier that appears on the **IOCONFIG** statement.
- If you code multiple **EDT** statements, you may omit the **ID** = parameter from only one of those statements. If you omit it from more than one statement, the MVS configuration program will perform a semantic check on the rest of the input statements and will then terminate. The I/O configuration will not be built.

The MVS configuration program assigns this identifier to the eligible device table that it creates from the **UNITNAME** statements that immediately follow this **EDT** statement. The next **EDT** statement encountered or the end of the input stream marks the end of this eligible device table definition.

Note: If you define only one eligible device table, IBM recommends that you assign it the same identifier as the I/O configuration data. This will simplify EDT selection for the operator during IPL.

DEVPREF = (*genericname*[,*genericname*]..)
Redefines the device order that MVS is to use when attempting to satisfy an allocation request. MVS attempts to allocate devices from each generic type in the order you have listed them, left-to-right, until the request is satisfied.

If MVS cannot satisfy the request using the specified order, MVS continues the attempt by using the order defined in the IBM-supplied device preference table.

As MVS goes through the device preference table attempting to allocate devices, it might encounter generic device types that you specified on the DEVPREF parameter. Because MVS already attempted, unsuccessfully, to allocate devices from that generic device type, it now bypasses that type and goes on to the next generic device type in the table.

The variable *genericname* must be a valid generic name. It may be an IBM-defined generic name or an installation-defined generic name. IBM-defined generic names are listed in Appendix A, "Generic Device Types" on page A-1. Installation-defined generic names are defined in the UIMs that your installation provides.

If you specify only one generic name, you can omit the parentheses.

Default:

- The MVS configuration program uses the IBM-supplied device preference table without additions or changes.

Required/Optional

- If only one eligible device table is to be defined, you can omit the EDT statement. If you do omit it, the MVS configuration program assigns the eligible device table the ID that appears on the **IOCONFIG** statement.
- If you are defining multiple eligible device tables on one invocation of the MVS configuration program, this statement is required. You must include one statement for each eligible device table you wish to define.

Placement

- If there is only one **EDT** statement, place it ahead of all **UNITNAME** statements.
- If there are multiple **EDT** statements, place each one immediately ahead of the **UNITNAME** statements that define its related eligible device table.

Example 1

This example shows a partial input stream. The installation has omitted an **EDT** statement from the stream.

The MVS configuration program will use all of the **UNITNAME** statements to create a single eligible device table. This table will have an ID of AB by default. The device preference table will be unchanged.

```
IOCONFIG ID=AB
IODEVICE .....
IODEVICE .....
   .
   .
UNITNAME .....
UNITNAME .....
   .
   .
NIPCON .....
```

Example 2

This example shows a partial input stream. From these statements, the MVS configuration program will create two eligible device tables.

The first table will contain the esoteric names specified on the two **UNITNAME** statements that immediately follow the first **EDT** statement. This table has an ID of 01. Whenever the installation uses this eligible device table, the IBM-supplied device preference table will be used unchanged.

The second eligible device table will contain the esoteric names specified on the three **UNITNAME** statements that follow the second **EDT** statement. This table will have an ID of 02. Whenever the installation uses this table, the device preference table will be changed so that the 3330 and the 3340 appear first.

```
IOCONFIG ID=99
IODEVICE .....
IODEVICE .....
   .
   .
EDT ID=01
UNITNAME .....
UNITNAME .....
EDT ID=02,DEVPREF=(3330,3340)
UNITNAME .....
UNITNAME .....
UNITNAME .....
NIPCON .....
```

IOCONFIG Statement

Use the **IOCONFIG** statement to assign an identifier to the I/O configuration data.

Define the eligible device table (EDT) for your system. Use the EDT statement type in the SCHEDxx member of the SYS1.PARMLIB. (See SPL: *Initialization and Tuning*.)

For MVS, the identifier distinguishes one I/O configuration from another. The system operator may use the identifier to select an I/O configuration while loading the system from the system control frame.

Rules for coding input statements are in "Coding Rules for Input Statements" on page 5-2.

[name]	**IOCONFIG**	**ID=***xx*

name
A 1-to-8 character name. The MVS configuration program ignores this name. You may wish to code it, however, to help you identify your input statements.

ID = *xx*
Assigns the identifier, *xx*, to the I/O configuration. The identifier *xx* can be any two-digit alphanumeric value. You can assign each digit a value 0-9 or A-Z.

Note: During IPL, if the operator allows the I/O configuration selection to default, the system will use the configuration that has the identifier 00.

Default: None

Required/Optional

This statement is required. Include one and only one **IOCONFIG** statement in the input stream.

If the **IOCONFIG** statement is omitted from the input stream or the input stream contains multiple **IOCONFIG** statements, the MVS configuration program will perform a semantic check of the input stream and will then terminate. The I/O configuration will not be built.

Placement

This statement may appear anywhere in the input stream. To provide an order to the input stream and to make it easier to read, however, you should place the **IOCONFIG** statement first.

Example

This example shows a partial input stream. From these statements, the MVS configuration program will create I/O configuration data. This data will have an ID of A1 and will include three eligible device tables. The first eligible device table will have an ID of 01, the second an ID of 02, and the third an ID of A1 (by default).

```
IOCONFIG ID=A1
IODEVICE .....
  .
  .
EDT ID=01
UNITNAME .....
  .
  .
EDT ID=02
UNITNAME .....
  .
  .
EDT
UNITNAME .....
  .
  .
NIPCON .....
```

IODEVICE Statement

Use IODEVICE statements to define the non-telecommunications devices, the telecommunications lines (this also defines the telecommunication terminals), and the telecommunication controllers in the I/O configuration.

Rules for coding input statements are in "Coding Rules for Input Statements" on page 5-2.

The parameter requirements for an IODEVICE statement vary according to the device. Therefore, before coding an IODEVICE statement refer to Chapter 6, "Defining I/O Devices" for specific device requirements. That chapter explains how to code an IODEVICE statement for each supported device.		
[*name*]	**IODEVICE**	**ADDRESS=**{ *device-number* | **i1.ADDRESS parameter** (*number-of-units*)} **CUNUMBR=**(*number*[,*number*]...) **,UNIT=***device-type* [,**ADDRESS=***adapter*] [,**FEATURE=**(*feature*[,*feature*]...)] [,**MODEL=***model-number*] [,**NUMSECT=**{*number*}] [,**OFFLINE=**{**YES**|**NO**}] [,**PATH=***chpid*] [,**PCU=***number*] [,**SETADDR=**{**0**|**1**|**2**|**3**}] [,**STADET=**{**Y**|**N**}] [,**TCU=**{**2701**|**2702**|**2703**}] [,**TIMEOUT=**{**Y**|**N**}] [,**UNITADD=***unit-address*]

ADAPTER = *adapter*

Specifies either the terminal control or transmission adapter used to connect a telecommunications line to a transmission control unit, or the type of channel adapter that connects a 3704 or 3705 communications controller to a channel path.

ADDRESS = {*device-number*|(*device-number*,*number-of-units*)}

Specifies the initial device number and how many device numbers you want the MVS configuration program to assign.

- The variable *device-number* is the initial device number.
- The variable *number-of-units* (also called the replication factor) specifies how many device numbers you want the MVS configuration program to assign.

device-number

Specifies the device number. For most devices, you can specify a hexadecimal value in the range 000 through FFF.

For some devices, there are limits on the values that you can specify for *device-number*. Before coding *device-number* for a device, see the IODEVICE statement description for that device in Chapter 6, "Defining I/O Devices." That is where the limits are explained.

- If you also code UNITADD= on the IODEVICE statement:
  - *device-number* can be any value that follows the rules, if any, for the device.
  - The sum of (*device-number*) plus (*number-of-units* - 1) must not exceed X'FFF'. (Before doing this calculation, convert *number-of-units* to a hexadecimal value.)
  - UNITADD= must specify the device's physical unit address.
- If you do not code UNITADD= on the IODEVICE statement:
  - The two rightmost digits of the *device-number* must specify the device's physical unit address.
  - The leftmost digit can be any hexadecimal value you choose.
  - The sum of (the two rightmost digits of *device-number*) plus (*number-of-units* - 1) must not exceed X'FF'. (Before doing this calculation, convert *number-of-units* to a hexadecimal value.)

Unless the device description in Chapter 6, "Defining I/O Devices" states otherwise, the MVS configuration program assigns device numbers in sequence starting with the device number you specify. For example:

- If you code ADDRESS=(000) or ADDRESS=(000,1), the MVS configuration program assigns one device number, 000.
- If you code ADDRESS=(000,4), the MVS configuration program assigns four device numbers, 000, 001, 002, and 003.

Note: Do not specify the same device number for more than one device in an I/O configuration.

number-of-units

Specifies how many device numbers the MVS configuration program is to assign. For a multiple exposure device, *number-of-units* specifies how many base exposures the MVS configuration program is to assign. You can specify a decimal number in the range 1 through 4095.

The MVS configuration program builds one UCB for each device number. For a multiple exposure device, this means one UCB for each exposure.

If you omit *number-of-units*, the MVS configuration program usually uses the default value of 1. Exceptions are noted in the individual device descriptions in Chapter 6, "Defining I/O Devices" on page 6-1.

IOCP assumes a *number-of-units* value of 1 for some devices. IBM, therefore, recommends that you explicitly specify *number-of-units* to avoid a possible conflict with the IOCP specification. See *IOCP User's Guide and Reference*.

CUNUMBR = (*number*[,*number*]...)
Specifies the control unit number(s) of the control units to which the device is attached. The number(s) you specify must also be specified on the CUNUMBR= parameter of the CNTLUNIT statement.

Specify 4 hexadecimal digits in the range 0000 through FFFF for each control unit.

The MVS configuration program does not use this parameter. For information about IOCP's use of the parameter, see the IOCP manual for the processor on which you will use the I/O configuration.

FEATURE = (*feature*[,*feature*]...)
Specifies the features available on the device. You may specify the features in any order.

For information about the features that you can specify for a specific device, see the description for that device in Chapter 6, "Defining I/O Devices" on page 6-1.

MODEL = *model-number*
Specifies the model number, if any, of the device. You can specify one or two alphanumeric characters for the model-number.

NUMSECT = {*number*}
Specifies the number of guaranteed 256-byte buffer sections in the 2840 display-control buffer to be allocated to a 2250 Model 3. This parameter is valid for a 2250 Model 3 only.

- If you omit the NUMSECT parameter, the available buffer space is divided equally among the 2250-3s. Any remaining buffer space is allocated to the last 2250-3 assigned.
- If you code the NUMSECT parameter, the buffer space is allocated as you specify. You must ensure that each 2250-3 that is attached to the control unit has at least one buffer allocated.

The total number of buffer sections guaranteed to the devices attached to a 2840 must not exceed the number of sections in the buffer of that 2840.

The device that you are specifying is the only device that can use these buffer sections.

For help with selecting an appropriate value for this parameter, see:

- *Introduction to the IBM 3250 Graphics Display System*
- *OS/VS Graphic Programming Support (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System*

OFFLINE = {YES|NO}
Specifies whether MVS is to consider the device online or offline at IPL.

To specify the device as offline, code OFFLINE = YES.

To specify the device as online, code OFFLINE = NO, or omit the parameter.

If MVS needs the device during IPL, specify OFFLINE = NO.

Note: Even if you specify **OFFLINE = NO** or omit the parameter for a device, the system still regards it as offline if the device is neither ready nor currently attached to the system at IPL.

PATH = *chpid*
Specifies the preferred channel path. The I/O subsystem uses the preferred channel path, if available, for the initiation of I/O operations. If the preferred path is busy or not available, any other available path will be used.

The MVS configuration program does not use this parameter. For IOCP the parameter is optional. For information about IOCP's use of the parameter, see the IOCP manual for the processor on which you will use the I/O configuration.

PCU = *number*
Identifies the 2840 model 2 control unit to which the 2250 model 3 is attached. The variable *number* may be any decimal value in the range 1 to 4095. The operating system uses this number to identify the 2250-3s that are attached to a given 2840-2 control unit.

Each 2840 can accept up to four 2250s. For a 2250-3 attached to a given control unit, you can specify any number in the range, 1 to 4095. For all 2250-3s that are attached to that same control unit, you must specify the same number, however. Once you have used a number to identify 2250-3s attached to a particular 2840-2, do not use that same number to identify 2250-3s that are attached to a different 2840-2.

For example, assume your configuration includes two 2840-2 control units. For all of the 2250-3s attached to one of these control units, you could specify **PCU = 1**. For all of the 2250-3s attached to the other control unit, you could specify **PCU = 2**.

Note: *Do not* code a separate IODEVICE statement for the 2840. That is, do not code an IODEVICE statement that specifies **UNIT = 2840**.

SETADDR = {0|1|2|3}
Specifies which of the 4 set address (SAD) commands is to be issued to the transmission control unit (TCU) for operations on the line specified by the ADDRESS operand.

Code SETADDR only if both of the following statements are true:

- The IODEVICE statement also specifies TCU = 2702
- The device description in "Telecommunication Device Definitions" on page 6-67 tells you to code the SETADDR parameter

If the TCU is a 2701 or 2703, the SETADDR parameter is ignored.

STADET = {Y|N}
Specifies whether the illegal status detection facility is to be enabled or disabled in 370-XA mode. To enable the facility, specify Y; to disable it, specify N.

In S/370 mode, the system ignores this facility.

The MVS configuration program and some versions of IOCP do not use this parameter. For those versions of IOCP that use the parameter, the parameter is optional. To determine if your version of IOCP uses the parameter, see the IOCP manual for the processor on which you will use the I/O configuration.

TCU = {2701|2702|2703}
Specifies the transmission control unit for a telecommunications line.

If you code TCU=2702, you might also have to code the SETADDR parameter. To determine if you have to code the SETADDR parameter, see the description of the telecommunication device you are defining. The device description is in Chapter 6, "Defining I/O Devices" on page 6-1.

Note: *Do not* code a separate IODEVICE statement for a 2701, 2702, or 2703. That is, do not code an IODEVICE statement that specifies **UNIT=2701**, **UNIT=2702**, or **UNIT=2703**.

TIMEOUT = {Y|N}
Specifies whether the I/O interface timeout function is to be active for channel-initiated I/O operations to the device. If the timeout function is active and a timeout condition occurs, the channel subsystem terminates the I/O operation to the device and generates an interface-control-check interruption.

Y Specifies that the timeout function is to be active. When Y is specified, the channel times I/O operations.

N Specifies that the timeout function is to be inactive (timeout is inhibited). The channel does not time I/O operations.

The MVS configuration program does not use this parameter. For IOCP this parameter is optional. For information about IOCP's use of the parameter, see the IOCP manual for the processor on which you will use the I/O configuration.

UNIT = *device-type*
Specifies the device type. For telecommunication lines, specifies the device type that is connected to the line. You can specify up to five alphanumeric characters for the device-type.

When you define a device that the MVS configuration program supports, specify the device-type shown in the applicable device description in Chapter 6, "Defining I/O Devices."

To define an unsupported device as a DUMMY, specify **UNIT=DUMMY**.

To define a device for which your installation has provided a unit information module (UIM), specify the device type required by the UIM design. For information about defining an unsupported device, see "Defining an Unsupported Device" on page 6-4.

UNITADD = *unit-address*
Specifies the device's physical unit address when the **ADDRESS=** parameter specifies a device number that does not include the physical unit address.

The variable *unit-address* must be the device's physical unit address and must be a hexadecimal value in the range X'00' to X'FF'.

For additional information about the UNITADD parameter, see the IOCP book that pertains to the processor on which you will use the I/O configuration.

Required/Optional

This statement is required for each uniquely addressable device and telecommunications line you want included in the I/O configuration. If you omit the IODEVICE statement for a particular device or telecommunications line, the MVS configuration program does not include that device or line in the I/O configuration.

Placement

You can place these statements anywhere in the input stream. IBM recommends, however, that you place all IODEVICE statements together immediately after the last CNTLUNIT statement.

Example 1

This statement defines a 3277 Model 2 with a device number of 009.

```
C009   IODEVICE   UNIT=3277,ADDRESS=009,MODEL=2
```

Example 2

This statement defines a 2540 Model 1 card punch with the CARDIMAGE feature. The device number for the device is 00D.

```
P00D   IODEVICE   UNIT=2540P,ADDRESS=00D,                    X
                FEATURE=CARDIMAGE,MODEL=1
```

Example 3

This statement defines a 3350 with 6 drives. The MVS configuration program assigns six device numbers 130, 131, 132, 133, 134, and 135.

```
D3350   IODEVICE   UNIT=3350,ADDRESS=(130,6)
```

Example 4

This statement specifies a 3330 Model 1. The MVS configuration program assigns device numbers 290 and 291 (The 3330 defaults to two device numbers).

```
IODEV   IODEVICE   UNIT=3330,MODEL=1,ADDRESS=290
```

Example 5

This statement specifies a 3350 attached to a 3880 Model 11 (also called a 3350P). The MVS configuration program assigns eight device numbers: the base device numbers are 2C4 and 2C5. The other assigned device numbers are 2CC, 2CD, 2D4, 2D5, 2DC, and 2DD.

```
  PAGDS  IODEVICE  UNIT=3350P,ADDRESS=(2C4,2)
```

Example 6

This statement specifies a 3350 attached to a 3880 Model 21 (also called a 3351P). The MVS configuration program assign eight device numbers: the base device numbers are 2C0 and 2C1. The other assigned device numbers are 2C4, 2C5, 2C8, 2C9, 2CC, and 2CD.

```
  PAGST  IODEVICE  UNIT=3351P,ADDRESS=(2C0,2)
```

NIPCON Statement

Use the **NIPCON** statement to identify devices that you will allow NIP to use as consoles. If you wish, you can code multiple **NIPCON** statements.

Rules for coding input statements are in "Coding Rules for Input Statements" on page 5-2.

[name]	**NIPCON**	**DEVNUM=**(*device-number*[,*device-number*]...)

name
A 1-to-8 character name. The MVS configuration program ignores this name. You may wish to code it, however, to help you identify your input statements.

DEVNUM = (*device-number*[,*device-number*]...)
The device-number of one or more devices that NIP can use as consoles. NIP tries to uses the devices in the order you list them, left-to-right.

The variable *device-number* must be a hexadecimal value in the range 000 to FFF. This device number must also be specified on an **IODEVICE** statement or fall within the range of device numbers specified on an **IODEVICE** statement.

If you specify only one device number, the parentheses are optional.

Default: None

Devices That You Can Define as NIP Consoles

The following chart lists devices that you can define as NIP consoles.

Figure 5-1. Devices That You Can Define as NIP Consoles

Device and model	Description
2740-1	Communications terminal - The terminal *must* have the CHECKING and OIU features and you *must* specify these features.
3270-X	Any device that can be defined as a 3270-X. NIP uses this device as though it were a 3277-2.
3277-2	Display station
3278-2, -2A,- 3, - 4, or -5	Display station
3279-2A, -2B, -2C, -3A, or -3B	Display station

Notes:

1. If you identify multiple devices as NIP consoles, MVS attempts to use them in the order they appear on the **NIPCON** statements. If you use multiple **NIPCON** statements, MVS attempts to use the devices listed on the first statement, then those listed on the second statement, and so forth.
2. To ensure that all system messages appear on one console, IBM recommends that you identify the master console as the first console on the first **NIPCON** statement.
3. All of the devices that you identify must be specified as being on-line: on the IODEVICE statement specify OFFLINE=NO or omit the OFFLINE parameter.

Required/Optional

This statement is required. You can include multiple **NIPCON** statements in the input stream.

If you omit **NIPCON** statements from the input stream or you fail to specify one of the devices as being on-line, The MVS configuration program does a syntax check and a semantics check on the input stream and then terminates without building the I/O configuration.

Placement

This statement may appear anywhere in the input stream. To provide an order to the input stream and to make it easier to read, however, IBM recommends that you place all **NIPCON** statements together at the end of the input stream.

Example

This example shows a partial input stream. The **IODEVICE** statements define devices that have device numbers 350, 421, and 450. The **NIPCON** statements identify these devices as consoles that NIP can use.

In this example, assume that the installation has defined the 3270-X with the device number 450 as the master console. Therefore, to ensure that all system messages appear on one console, 450 should be the first device number specified on the first **NIPCON** statement.

MVS attempts to use the devices in the order they are defined: 450, 350, and 421.

```
IOCONFIG ID=BB
IODEVICE ADDRESS=421,UNIT=3270,MODEL=X,.....
IODEVICE ADDRESS=350,UNIT=3277,MODEL=1,.....
IODEVICE ADDRESS=450,UNIT=3270,MODEL=X,.....
  .
  .
EDT ID=01
UNITNAME .....
UNITNAME .....
  .
  .
  .
NIPCON DEVNUM=(450,350)
NIPCON DEVNUM=421
```

UNITNAME Statement

Use the UNITNAME statement to:

- Define an esoteric device group, assign the group a name, and indicate whether the group is to be eligible for VIO
- Identify generic device types that you want to be eligible for VIO

Specification Rules

Code UNITNAME statements according to the following rules:

- The device numbers specified must also be specified on an IODEVICE statement.
- If NAME= specifies a generic name, do not code the UNIT= parameter.
- You can specify VIO=YES only for:
  - An esoteric device group that contains at least one DASD
  - A generic device type that contains DASD
- Do not use a UNITNAME statement to assign a name to a group of telecommunication devices. If you do, when the assigned name is used on a DD statement, MVS will allocate only the first device specified on the UNIT= parameter.
- You can code a maximum of 255 characters in the operand field of a UNITNAME statement. If more device numbers must be listed than can fit on one statement, additional UNITNAME statements can be coded using the same variable in the NAME= parameter. Those UNITNAME statements with the same NAME= variable must be grouped contiguously in the input stream. If they are separated, the MVS configuration program assumes that the device numbers on the first UNITNAME statement are the only device numbers that belong to that device group.
- The MVS configuration program *does not* limit the number of esoteric device groups that you can define (the SYSGEN process limited you to 256).
- You do not have to assign esoteric names to devices that you define as UNIT=DUMMY. The MVS configuration program assigns these devices a generic name of DUMMY. Users can specify the generic name DUMMY for allocation purposes.
- If you use the catalog from the previous generation, and that catalog has esoteric names for the data sets, put the UNITNAME macros in the same sequence, with the same device classes, as in the previous generation. Make no deletions. Make additions only at the end.

Rules for coding input statements are in "Coding Rules for Input Statements" on page 5-2.

[*name*]	**UNITNAME**	**NAME=**{*esoteric-group-name* | *generic-name*} [,**UNIT=**{*device-number* | (*device-number* [,*device-number*]...) | ((*device-number*,*n*)) | ((*device-number*,*n*) [,(*device-number*,*n*)]...)}] [,**VIO=**{**YES**|**NO**}]

NAME= {*esoteric-group-name*|*generic-name*}

To define an esoteric device group, specify **NAME=** *esoteric-group-name*. The variable, *esoteric-group-name* is the name you want assigned to the group. The variable *esoteric-group-name* must be from 1 to 8 characters. These characters can be alphameric, national (#, @, or $), or the 2 special characters, slash (/) and hyphen (-).

To specify a generic device type that is eligible for VIO, specify **NAME=***generic-name*. NAME=*generic-name* must specify a DASD device type, and you must also specify **VIO=YES**. The variable *generic-name* must be either an IBM-defined generic name listed in Appendix A, "Generic Device Types" on page A-1, file or an installation-defined generic name.

Unless you are specifying a generic device type that is eligible for VIO, do not supply a **UNITNAME** statement to define a generic device type; MVSCP automatically generates generic entries for unit types specified in IODEVICE statements.

CAUTION:

- **Do not specify NAME=SYSALLDA. This name is reserved for use by MVS only.**
- **If the I/O configuration includes the Mass Storage Subsystem (MSS), the MSS Table Create Program generates UNITNAME statements. Be sure that the variables that you specify on NAME= are different than those generated by the MSS.**

  For additional information about the MSS Table Create Program, see *OS/VS Mass Storage Control Table Create.*

UNIT=

Specifies the device numbers of a group of I/O devices. The group can consist of one or more devices. The name you code on the NAME= parameter becomes the name of the group.

Notes:

1. Code the UNIT= parameter only if the NAME= parameter specifies an esoteric group name.
2. IBM recommends that all devices you assign to an esoteric device group be members of the same device class with the following exception: you may define a group that contains both DASD and magnetic tape devices.

device-number
Specifies the device number of an I/O device. You can specify 3 hexadecimal digits, 000 through FFF.

The device number must also be defined on the ADDRESS= parameter of an IODEVICE statement.

For a multiple exposure device (2305-2, 3350P or 3351P), *device-number* must be a base exposure.

- For a 2305-2, the base exposure is the device number that appears on the ADDRESS= parameter of the IODEVICE statement.
- For the 3350P or the 3351P, see Appendix C, "Determining 3350P and 3351P Device Number Assignments" on page C-1 for a list of base exposures.

If you specify a non-base exposure, the MVS configuration program ignores the specification and issues an informational message, but otherwise, runs as though the non-base exposure had not been specified.

((*device-number,n*))
Specifies a sequence of consecutive device numbers. The variable *device-number* is the first device number in the sequence. The variable *n*, a 1- to 3-digit decimal number, indicates how many device numbers to include in the sequence. The default for *n* is 1.

For example:

- UNIT=((160,2)) specifies device numbers 160 and 161. This is equivalent to coding UNIT=(160,161)
- UNIT=((200,4)) specifies device numbers 200, 201, 202, and 203. This is equivalent to coding UNIT=(200,201,202,203)

Each of the device numbers in the sequence must also be defined on the ADDRESS= parameter of an IODEVICE statement.

For a multiple exposure device (2305-2, 3350P or 3351P), the sequence of specified device numbers must include base exposures only.

- For a 2305-2, the base exposure is the device number that appears on the ADDRESS= parameter of the IODEVICE statement.
- For the 3350P or the 3351P, see Appendix C, "Determining 3350P and 3351P Device Number Assignments" on page C-1 for a list of base exposures.

If the sequence specifies a non-base exposure, the MVS configuration program ignores the specification and issues an informational message, but otherwise, works normally.

You may mix the two forms of the UNIT parameter. If more than one value is expressed, the values must be enclosed in parentheses. If the form (device number, n) is used as the only subparameter on the statement, use double parentheses. For example, UNIT=((180,4)) specifies a group of 4 devices that have the device numbers 180, 181, 182, and 183.

Note: If you use the form UNIT=((*device-number,n*)), be sure that the sum of (*device-number*) plus (*n* - 1) does not exceed X'FFF'. (To do this calculation, you must convert *n* to a hexadecimal value.)

VIO = {YES|NO}
Specifies whether or not you want this group of devices to be eligible for VIO.

You can specify VIO = YES only for:

- An esoteric device group that contains at least one DASD
- A generic device type that contains DASDs

For information on VIO, see *Initialization and Tuning*.

Required/Optional

UNITNAME statements are required if:

- You want to define esoteric device groups
- You want to make devices eligible for VIO

Note: The cataloged procedures that your installation uses determine which esoteric device groups you must define. For example, if you use certain procedures from the IBM-provided SYS1.PROCLIB, you will have to define esoteric device groups SYSDA and SYSSQ. To determine which esoteric device groups you must define, review your cataloged procedures.

Placement

If the input stream contains no EDT statement, IBM recommends that you place the UNITNAME statements immediately after the last IODEVICE statement.

If the input stream contains one or more EDT statements, you must place the UNITNAME statements immediately after the EDT statement to which they correspond.

Example 1

This example defines an esoteric device group and names it TAPE. The group includes the devices that have device numbers 180, 181, 280, 281, and 282.

```
UNITNAM1   UNITNAME   NAME=TAPE,                          X
               UNIT=(180,181,280,281,282)
```

Example 2

This example defines an esoteric device group and names it DISK. The group includes the devices that have device numbers 130, 131, 132, 133, 134 135, 136, and 137.

```
UNITNAM2   UNITNAME   NAME=DISK,UNIT=((130,8))
```

Example 3

This example assigns the esoteric name 167 to an unsupported device. The unsupported device was assigned device number 167 (specified as UNIT = DUMMY,ADDRESS = 167 on an IODEVICE statement).

```
UNITNAM3   UNITNAME   NAME=167,UNIT=167
```

Example 4

This example defined an esoteric device group and names it SYSDA. The group is eligible for VIO.

```
UNITNAM4   UNITNAME   NAME=SYSDA,VIO=YES,                 X
               UNIT=((110,2),(150,8),(213,5),              X
               (220,8),(250,8),(350,8),(410,8))
```

Chapter 6. Defining I/O Devices

The MVS configuration program supports many IBM-built devices and several non-IBM devices. This book calls a device that the MVS configuration program supports a *supported device*. A device the MVS configuration program does not support is called an *unsupported device*.

For each supported device, the MVS configuration program provides a program called a unit information module (UIM). It is the UIM that recognizes and processes the device's IODEVICE statement. An unsupported device is one for which the MVS configuration program provides no UIM.

Even though the MVS configuration program does not support a particular device, you can still define and use that device. How to do this is explained later in the topic "Defining an Unsupported Device" on page 6-4. Before you define any device, however, you must first determine if the MVS configuration program does or does not support the device.

In a system with stringent security requirements, you might need to ensure that only certain programs can allocate unit record, communication, or graphics devices. If your installation has such security requirements, see "Providing Additional Security for Devices" on page 6-6.

Locating Information in this Chapter

This chapter contains the following major topics:

Determining if The MVS Configuration Program Supports a Device

With the exception of telecommunication binary synchronous communication (BSC) terminals, all supported devices are listed in the index. To determine if the MVS configuration program supports a particular device, look for the device type in the index under the entry titled "Devices Supported," or look in the index under the specific device type. If you find the device listed, the page reference refers you to the topic that explains how to code an IODEVICE statement for that particular device.

If the index does not list the device type, the MVS configuration program does not support the device. To define an unsupported device, see the topic "Defining an Unsupported Device" on page 6-4.

Note: The MVS configuration program supports all of the devices that the MVS/XA SYSGEN process supported except the 2260.

The index does not list individual BSC (binary synchronous communication) terminals. (A BSC terminal is any device that supports the BSC protocol.) Instead, the index lists the type of communications line to which the BSC terminals attach. The types of communication lines are:

- switched point-to-point
- nonswitched point-to-point
- nonswitched multipoint.

The index entries for these line types direct you to the topic that explains how to code the IODEVICE statement that defines the line. By defining the line you also define the BSC terminals that are attached to the line.

To determine if a particular device supports the BSC protocol, see the hardware manual for that device.

Defining a Supported Device

To define a supported device, you need only code an IODEVICE statement. There are several ways to find the topic that explains how to code the IODEVICE statement:

- Look in the index for the specific device type, such as 2250-3, 3350, or 4248.
- Look in the index under the entry titled "devices supported."
- Refer to the earlier topic in this chapter "Locating Information in this Chapter" on page 6-1. That topic defines where, in this chapter, the IODEVICE statements are described for the different device classes. Once you have located the device class within this chapter, the devices are listed in ascending alphanumerical order.

Defining an Unsupported Device

To define an unsupported device, do one of the following:

- Define the device as a DUMMY device
- Provide a unit information module (UIM) that supports the device

Defining a DUMMY Device

The MVS configuration program provides a UIM that recognizes any device that you define as a DUMMY device. The MVS configuration program treats each DUMMY device as though it were a unit record device. That is, the MVS configuration program sets the UCBTYP field in the device's UCB to X'00000800'. The MVS configuration program also assigns each DUMMY device a generic name of DUMMY and an error routine value of X'00' (the system converts the value X'00' to an error recovery program name, IGE0001C.

Note: The MVS/XA SYSGEN process allowed you to define both the UCBTYP field and the error routine value. To do this, you coded the DEVTYPE parameter and the ERRTAB parameter on the IODEVICE statement. If these parameters appear on an IODEVICE statement in the MVS configuration program input stream, the MVS configuration program ignores them.

To define a DUMMY device, all you need to do is code the appropriate IODEVICE statement. For specific information about coding this IODEVICE statement, see "DUMMY Device" on page 6-55.

The following example shows an IODEVICE statement that defines a DUMMY device.

```
IODEVICE UNIT=DUMMY,ADDRESS=010,CUNUMBR=001
```

Writing a UIM to Support a Device

To define an unsupported device as a device type other than DUMMY, you must provide a UIM. You must also code an IODEVICE statement for the device.

The UIM must recognize and process the IODEVICE statement that you code. The UIM design determines which parameters can be coded on the IODEVICE statement.

IBM has provided a sample UIM in SYS1.SAMPLIB member SAMPUIM. You can modify this sample to create your own UIM. Detailed information about coding a UIM is in *System Modifications*.

Other Products May be Required

There are some devices that cannot be defined or used until an additional product is installed. The product must be installed on both the system where you define the device (that is the system where you run the MVS configuration program) and on the system where you use the device. For example, the 3851 and the 3330V cannot be defined or used unless you have MSS installed.

> **Note:** If you try to use an I/O configuration that includes one of these devices, and the system where you use the configuration does not include the required product, MVS will enter wait state during IPL processing.

Devices that require additional products and the products they require are listed in the following table. For a list of each products FMID, see the program directory.

Device	Required Product
1287	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
1288	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
2250	Graphics Access Method/SP2 (GAM/SP2)
3330V	Mass Storage System Extensions with the XA Feature
3540	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
3838	Vector Processing Subsystem/Extended Architecture (VPSS/XA)
3848	Cryptographic Unit Support
3851	Mass Storage System Extensions with the XA Feature
3886	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
3890	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
3895	Magnetic Ink Character Reader/Optical Character Reader (MICR/OCR)
5080	Graphics Access Method/SP2 (GAM/SP2)

Providing Additional Security for Devices

If your system has stringent security requirements and includes RACF 1.9, you can ensure that only certain programs can allocate unit record, communication, or graphics devices. These programs include PSF (Print Services Facility) for printers, ACF/VTAM (Advanced Communication/Virtual Telecommunications Access Method) for communication or graphics devices, and JES2 or JES3 for unit record, communication, or graphics devices.

To allow this control, allocation uses SAF to call RACF to determine if the user has authority to access the device. The connection between allocation and RACF is either the esoteric or generic device name. When a user attempts to allocate a device, the system passes the device name to RACF. RACF uses the device name to locate a profile and determine whether the user can access the device. If the user does not have authority to access the device, the allocation fails. (Note that the system does not retry an allocation request that fails because the user is not authorized to access the device.)

Work with your RACF security administrator while planning additional device security. Follow this procedure:

1. Determine your exact security requirements according to your applications. Consider questions such as these:
   - Are there some devices that only a few users can use?
   - Are there some devices that all users can use?
   - Do some devices share the same security requirements?
2. Assign a specific device name to each device that has a unique security requirement. Assign a generic device name to each device group that shares security requirements.
3. Give the device names to your RACF security administrator.

For more details, see the *RACF Security Administrator's Guide*.

Channel-to-Channel Adapter Definitions

This topic explains how to code IODEVICE statements to define a channel-to-channel (CTC) adapter or a 3088. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending alphanumerical order by device type.

You must code one IODEVICE statement for *each* channel-to-channel attachment in your configuration. A channel-to-channel attachment is the addressable path through a CTC adapter or through a 3088 that connects your system to another system.

CTC Adapter or 3088

- Code an IODEVICE statement for each channel-to-channel attachment in the configuration.
- *Do not* connect multiple systems through multiple CTCs or multiple 3088s that are on the same channel path. With this type of a connection, channel path contention can cause a lockout condition.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **CTC**

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The adapter or the 3088 is attached to a System 370.	**370**

DASD Definition

This topic explains how to code IODEVICE statements for supported direct access storage devices. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending numerical order by device type. The heading at the top of each page identifies the class of device, DASD, the device type, and model.

2305 Model 2

The 2305-2 is a multiple exposure device.

Required Parameters

ADDRESS = (*device-number***,1)**

Notes:

1. *device-number* must be a value in the range '000'X to 'FF8'X.
2. The MVS configuration program assigns eight (8) consecutive device numbers starting with the device number you specify. The device number that you specify is called the 2305's base exposure. The other seven device numbers that the MVS configuration program assigns are called non-base exposures. Of the eight device numbers, the base exposure is the only one you can include on the UNIT= parameter of a UNITNAME statement.

CUNUMBR = (*number*[,*number*]...**)**
MODEL = 2
UNIT = 2305

Optional Parameters

FEATURE = (*feature*[,*feature*]...**)** (See "Device Features")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3330 Model 1 or 2

To define a 3330 as an MSS virtual volume (**UNIT=3330V**), see, "MSS Device Definitions" on page 6-64.

Required Parameters

ADDRESS = {*device-number* | (*device-number,number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3330**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

MODEL = **1**

Notes:

1. The default is model 1.
2. You must define the model 2 as a model 1.

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3330 Model 11

Required Parameters

ADDRESS = {*device-number* | (*device-number*,*number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
MODEL = **11**
UNIT = **3330**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3333 Model 1

The 3333-1 is functionally equivalent to the 3330-1.

Required Parameters

ADDRESS = {*device-number* | (*device-number*,*number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3330**

Note: You must define the 3333 as a 3330.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

MODEL = **1**

Note: The default is model 1.

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3333 Model 11

The 3333-11 is functionally equivalent to a 3330-11.

Required Parameters

ADDRESS = {*device-number* | (*device-number,number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
MODEL = **11**
UNIT = **3330**

Note: You must define the 3333 as a 3330.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3340

Required Parameters

ADDRESS = {*device-number* | (*device-number*,*number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3340**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
Rotational position sensing (Required for 3340 compatibility mode).	**RPS**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3344

The 3344 is functionally equivalent to four IBM 3340s mounted with a 3348 Model 70 head disk assembly. Each 3344 has two drives and requires eight logical addresses. For information on addressing requirements, see the appropriate control unit manual.

Code an IODEVICE statement for each 3340 that is to be simulated by the 3344.

Required Parameters

ADDRESS = {*device-number* | (*device-number,number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3340**

Note: You must define the 3344 as a 3340.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
Rotational position sensing (Required for 3340 compatibility mode).	**RPS**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3350

You can use the 3350 in either of two modes:

- 3350 native mode (the 3350 functions as a 3350)
- 3330 compatibility mode (the 3350 functions as either a 3330-1 or a 3330-11)

The topic you are reading explains how to code the IODEVICE statement so you can use the 3350 in native mode.

To use the 3350 in compatibility mode, code the IODEVICE statement as you would for either a 3330-1 or a 3330-11. For information on how to code these IODEVICE statements, see the appropriate topic:

- "3330 Model 1 or 2" on page 6-11
- "3330 Model 11" on page 6-12

For additional information about 3330 compatibility mode, see *3350/3344 Installation and Conversion Guide*.

Required Parameters

ADDRESS = {*device-number* | (*device-number,number-of-units*)}

Note: If you omit *number-of-units*, the MVS configuration program assigns two device numbers: the device number you specify and the next sequential device number.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3350**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3350 (Attached to 3880-11) or 3350P

- A 3350 that is attached to a 3880-11 is called a 3350P.
- When a user or programmer must specify device type, such as on a **DD** statement **UNIT=** parameter, they must specify the device type as 3350, not as 3350P. When you define the device on the **IODEVICE** statement, however, you must specify **UNIT=3350P**.
- The 3350P is a multiple exposure device.

Required Parameters

ADDRESS= {*device-number* | (*device-number*,*number-of-units*)}

Note: The variable *device-number* must be a 3-digit hexadecimal value in the range 000 to FE6. The variable *number-of-units* (also called the replication factor) must be a decimal value in the range 1 to 8 or may be omitted. If you omit *number-of-units* or specify 1, the MVS configuration program uses a default value of 2.

In order to explain the rules for coding the ADDRESS= parameter, the symbols *xyz* will represent the three digits of the device address and the symbol *r* will represent the number-of-units or replication factor. Using these symbols the ADDRESS= parameter can be shown as ADDRESS=(xyz,r). The ADDRESS= parameter must adhere to the following rules:

- x (the leftmost digit) must be in the range 0 to F.
- y (the middle digit) must be even (0,2,4,6,8,A,C, or E).
- z (the rightmost digit) must be less than 7 (0 to 6).
- The sum of z + r must be less than 9 (z + r < 9). For this calculation, if you omit *number-of-units* or code a value of 1, use a value of 2 for *r*.

The variable *number-of-units* determines how many device numbers the MVS configuration program assigns (it always assigns a minimum of eight). The MVS configuration program first assigns the device number you specify. It then uses that number as a base to calculate the additional device numbers that it assigns. The following chart shows, for different values of *number-of-units*, how many device numbers the MVS configuration program assigns.

If *number-of-units* is:	**The number of device numbers assigned is:**
omitted, 1, or 2	8
3	12
4	16
5	20
6	24
7	28
8	32

The actual values of the device numbers the MVS configuration program assigns depends on the value of the device number you specify. You must be aware of the assigned device numbers so you do not assign them to another device. For more information about the device numbers that the MVS configuration program assigns and the maximum allowable values for *number-of-units*, see Appendix C, "Determining 3350P and 3351P Device Number Assignments" on page C-1.

The only device numbers that you can code on the UNIT= parameter of a UNITNAME statement are the base exposures.

CUNUMBR=(*number*[,*number*]...**)**
UNIT=3350P

Optional Parameters

OFFLINE={**YES**|**NO**}
PATH=*chpid*
STADET={**Y**|**N**}
TIMEOUT={**Y**|**N**}
UNITADD=*nnn*

Device Features

None supported.

3350 (Attached to a 3880-21) or 3351P

- A 3350 that is attached to a 3880-21 is called a 3351P.
- When a user or programmer must specify device type, such as on a **DD** statement **UNIT=** parameter, they must specify the device type as 3350, not as 3351P. When you define the device on an **IODEVICE** statement, however, you must specify **UNIT=3351P**.
- The 3351P is a multiple exposure device.

Required Parameters

ADDRESS= {*device-number* | (*device-number,number-of-units*)}

Note: The variable *device-number* must be a 3-digit hexadecimal value in the range 000 to FF2. The variable *number-of-units* (also called the replication factor) must be a decimal value in the range 1 to 4 or may be omitted. If you omit *number-of-units* or specify 1, the MVS configuration program uses a default value of 2.

In order to explain the rules for coding the ADDRESS= parameter, the symbols *xyz* will represent the three digits of the device address and the symbol *r* will represent the number-of-units or replication factor. Using these symbols the ADDRESS= parameter can be shown as ADDRESS=(xyz,r). The ADDRESS= parameter must adhere to the following rules:

- x (the leftmost digit) must be in the range 0 to F.
- y (the middle digit) must be in the range 0 to F.
- z (the rightmost digit) must be less than 3 (0 to 2).
- The sum of z + r must be less than 5 (z + r < 5). For this calculation, if you omit *number-of-units* or code a value of 1, use a value of 2 for *r*.

The variable *number-of-units* determines how many device numbers the MVS configuration program assigns (It always assigns a minimum of eight). The MVS configuration program first assigns the device number you specify. The MVS configuration program then uses that number as a base to calculate the additional device numbers that it assigns. The following chart shows, for different values of *number-of-units*, how many device numbers the MVS configuration program assigns.

If *number-of-units* is:	**The number of device numbers assigned is:**
omitted, 1, or 2	8
3	12
4	16

The actual values of the device numbers the MVS configuration program assigns depends on the value of the device number you specify. You must be aware of the assigned device numbers so you do not assign them to another device. For more information about the device numbers that the MVS configuration program assigns and the maximum allowable values for *number-of-units*, see Appendix C, "Determining 3350P and 3351P Device Number Assignments" on page C-1.

The only device numbers that you can code on the UNIT= parameter of a UNITNAME statement are the base exposures.

CUNUMBR=(*number*[,*number*]...**)**
UNIT=3351P

Optional Parameters

OFFLINE={**YES**|**NO**}
PATH=*chpid*
STADET={**Y**|**N**}
TIMEOUT={**Y**|**N**}
UNITADD=*nnn*

Device Features

None supported.

3375

Required Parameters

ADDRESS = {*device-number* | (*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3375**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3380

Required Parameters

ADDRESS = {*device-number* |(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3380**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

This page intentionally left blank.

Display Device Definitions

This topic explains how to code IODEVICE statements for supported display devices. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending numerical order by device type. The heading at the top of each page identifies the class of device, display device, the device type, and model.

2250 Model 3

The system on which you run the MVS configuration program to define the 2250-3, and the system on which you use the 2250-3 must both have Graphics Access Method/SP2 (GAM/SP2) installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

For information that will help you select appropriate values for the parameters NUMSECT, and PCU, see the following books:

- *Introduction to the IBM 3250 Graphics Display System*
- *OS/VS Graphic Programming Support (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System*

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Notes:

1. The default for *number-of-units* is 1.
2. Do not assign this 2250 a device number that is within the range of device numbers of a 2250 that is attached to a different 2840 model 2 control unit.

CUNUMBR = (*number*[,*number*]....)
MODEL = **3**
PCU = *number*
UNIT = **2250**

Optional Parameters

FEATURE = (*feature*[,*feature*]..) (See "Device Features")
NUMSECT = *number*
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Alphameric keyboard (Required if you use the 2250 as an operator's console)	**ALKYB2250**
Programmed function keyboard	**PRGMKYBD**

3180 Model 1 Display Station

You can define a 3180-1 that is in 3278 compatibility mode as one of the following devices:

- 3270-X (See "3270 Model X" on page 6-31)
- 3278-1, -2, -2A, -3, or -4 (See "3278 Model 1, 2, 2A, 3, 4, or 5 Display Station" on page 6-36)

3250

The 3250 is functionally equivalent to a 2250-3 that is attached to a 2840-2.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = **3**
PCU = *number*
UNIT = **2250**

Note: You must define the 3250 as a 2250 model 3.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
NUMSECT = *number*
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Alphameric keyboard	**ALKYB2250**
Programmed function keyboard	**PRGMKYBD**

3251

The 3251 is functionally equivalent to a 2250 model 3. Therefore, you must define the 3251 as though it were a 2250 model 3. For instructions on how to code the IODEVICE statement, see "2250 Model 3" on page 6-26.

3268 Model 2 Printer

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = **2**
UNIT = **3286**

Note: You must define the 3268 model 2 as a 3286 model 2.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3268 is attached to a 3274-D control unit. This feature may improve the performance of the 3268 if programs that access the device use the appropriate channel programs.	**PTREAD**

3270 Model X

A 3270-X device is any device that has been defined on an IODEVICE statement as a 3270 Model X.

You can define any of the following as a 3270-X:

- 3180-1 that is in 3278 compatibility mode
- 3278-2, -3, -4, or -5
- 3279-S2B, -S3G, -2B, -2X, -3B, or -3X
- 3290
- Any device that supports the 3270 data stream and one of the following screen sizes. Screen size is shown as (row x column).

24 x 80	31 x 160	50 x 106
27 x 132	32 x 80	62 x 80
31 x 80	43 x 80	62 x 160

  The control unit to which the device is attached must be at a customization level that permits MVS to query the device. If MVS cannot query the device or cannot determine the actual device type, MVS treats the device as a 3277-2.

Any 3270-X device can be physically replaced by another 3270-X device without the need to change the I/O configuration definition (IODEVICE statement).

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = **X**
UNIT = **3270**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-32)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Character generator (Select only one)	
ASCII A	**ASCACHAR**
ASCII B	**ASCBCHAR**
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
Audible alarm (Requires a keyboard)	**AUDALRM**
Magnetic card reader adapter	**MAGCDRD**
Numeric lock	**NUMLOCK**
The 3270-X is attached to a 3274-D control unit. This feature may improve the performance of the 3270-X if programs that access the device use the appropriate channel programs.	**PTREAD**
Selector pen (light pen)	**SELPEN**
Keyboard type (Select only one)	
ASCII typewriter	**ASKY3277**
EBCDIC data entry	**DEKY3277**
EBCDIC typewriter	**EBKY3277**
78-key operator console	**OCKY3277**
Keyboard size (Select only one)	
70-key	**KB70KEY**
78-key	**KB78KEY**
81-key	**KB81KEY**

Required parameter combinations are shown in the next chart.

Required Parameter Combinations

If you specify	you must also specify:
KACHAR	**EBKY3277** and **KB81KEY** or **DEKY3277** and **KB70KEY**
KB70KEY	**DEKY3277** and **KACHAR**
KB78KEY	**ASKY3277** or **EBKY3277** or **OCKY3277**
KB81KEY	**EBKY3277** and **KACHAR**
AUDALRM	**ASKY3277** or **DEKY3277** or **EBKY3277** or **OCKY3277**

3274 Model 1A Control Unit

1. If each device that is attached to the 3274-1A has its own device number, do not code an IODEVICE statement for the 3274-1A. You must, instead, code an IODEVICE statement for each of the attached devices. For instructions on how to code the IODEVICE statements, see the appropriate device description in this chapter.
2. If the 3274-1A is locally attached and it has systems network architecture (SNA) support that permits the control of multiple devices through one device number, you must code an IODEVICE statement for the 3274-1A. Do not code IODEVICE statements for any of the devices attached to the 3274-1A.

   You must define the 3274-1A as though it were a 3791L. For instructions on how to code the IODEVICE statement, see "3791L Controller" on page 6-89.

3277 Model 1 or 2 Display Station

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {1|2}
UNIT = 3277

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-35)
OFFLINE = {YES|NO}
PATH = *chpid*
STADET = {Y|N}
TIMEOUT = {Y|N}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Character generator (Select only one)	
ASCII A	**ASCACHAR**
ASCII B	**ASCBCHAR**
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
Audible alarm (Requires a keyboard)	**AUDALRM**
Magnetic card reader adapter	**MAGCDRD**
Numeric lock	**NUMLOCK**
The 3277 is attached to a 3274-D control unit. This feature may improve the performance of the 3277 if programs that access the device use the appropriate channel programs.	**PTREAD**
Selector pen (light pen)	**SELPEN**
Keyboard type (Select only one)	
ASCII typewriter	**ASKY3277**
EBCDIC data entry	**DEKY3277**
EBCDIC typewriter	**EBKY3277**
78-key operator console	**OCKY3277**
Keyboard size (Select only one)	
70-key	**KB70KEY**
78-key	**KB78KEY**
81-key	**KB81KEY**

Required parameter combinations are shown in the next chart.

Required Parameter Combinations

If you specify	you must also specify:
KACHAR	**EBKY3277** and **KB81KEY** or **DEKY3277** and **KB70KEY**
KB70KEY	**DEKY3277** and **KACHAR**
KB78KEY	**ASKY3277** or **EBKY3277** or **OCKY3277**
KB81KEY	**EBKY3277** and **KACHAR**
AUDALRM	**ASKY3277** or **DEKY3277** or **EBKY3277** or **OCKY3277**

3278 Model 1, 2, 2A, 3, 4, or 5 Display Station

You can define models 2, 2A, 3, or 4 as either a 3278 or a 3270-X.

You *must* define a model 5 as a 3270-X.

You can define a 3180-1 that is in 3278 compatibility mode as a 3278-1, -2, -2A, -3, or, -4.

Any 3270-X device (a 3270-X device is any device that has been defined on an IODEVICE statement as a 3270 Model X) can be physically replaced by another 3270-X device without the need to change the I/O configuration definition (IODEVICE statement). "3270 Model X" on page 6-31 explains how to define a device as a 3270-X.

The remainder of this topic explains how to define a 3278 as a 3278.

Required Parameters

ADDRESS = {*device-number*|(*device-number*,*number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**|**2A**|**3**|**4**}
UNIT = **3278**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-37)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Character generator (Select only one)	
ASCII A	**ASCACHAR**
ASCII B	**ASCBCHAR**
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
Audible alarm (Requires a keyboard)	**AUDALRM**
Magnetic card reader adapter	**MAGCDRD**
Numeric lock	**NUMLOCK**
The 3278 is attached to a 3274-D control unit. This feature may improve the performance of the 3278 if programs that access the device use the appropriate channel programs.	**PTREAD**
Selector pen (light pen)	**SELPEN**
Keyboard type (Select only one)	
ASCII typewriter	**ASKY3277**
EBCDIC data entry	**DEKY3277**
EBCDIC typewriter	**EBKY3277**
78-key operator console	**OCKY3277**
Keyboard size (Select only one)	
70-key	**KB70KEY**
78-key	**KB78KEY**
81-key	**KB81KEY**

Required parameter combinations are shown in the next chart.

Required Parameter Combinations

If you specify	you must also specify:
KACHAR	**EBKY3277** and **KB81KEY** or **DEKY3277** and **KB70KEY**
KB70KEY	**DEKY3277** and **KACHAR**
KB78KEY	**ASKY3277** or **EBKY3277** or **OCKY3277**
KB81KEY	**EBKY3277** and **KACHAR**
AUDALRM	**ASKY3277** or **DEKY3277** or **EBKY3277** or **OCKY3277**

3279 Models S2B, S3G, 2A, 2B, 2C, 2X, 3A, 3B, and 3X

You can define models 2B or 3B as either a 3279 or a 3270-X.

You can define the following models only as a 3270-X: S2B, S3G, 2X, and 3X.

Any 3270-X device (a 3270-X device is any device that has been defined on an IODEVICE statement as a 3270 Model X) can be physically replaced by another 3270-X device without the need to change the I/O configuration definition (IODEVICE statement). "3270 Model X" on page 6-31 explains how to define a device as a 3270-X.

The reminder of this topic explains how to define a 3279 as a 3279.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**2A**|**2B**|**2C**|**3A**|**3B**}
UNIT = **3279**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-39)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Character generator (Select only one)	
ASCII A	**ASCACHAR**
ASCII B	**ASCBCHAR**
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
Audible alarm (Requires a keyboard)	**AUDALRM**
Magnetic card reader adapter	**MAGCDRD**
Numeric lock	**NUMLOCK**
The 3279 is attached to a 3274-D control unit. This feature may improve the performance of the 3279 if programs that access the device use the appropriate channel programs.	**PTREAD**
Selector pen (light pen)	**SELPEN**
Keyboard type (Select only one)	
ASCII typewriter	**ASKY3277**
EBCDIC data entry	**DEKY3277**
EBCDIC typewriter	**EBKY3277**
78-key operator console	**OCKY3277**
Keyboard size (Select only one)	
70-key	**KB70KEY**
78-key	**KB78KEY**
81-key	**KB81KEY**

Required parameter combinations are shown in the next chart.

Required Parameter Combinations

If you specify	you must also specify:
KACHAR	**EBKY3277** and **KB81KEY** or **DEKY3277** and **KB70KEY**
KB70KEY	**DEKY3277** and **KACHAR**
KB78KEY	**ASKY3277** or **EBKY3277** or **OCKY3277**
KB81KEY	**EBKY3277** and **KACHAR**
AUDALRM	**ASKY3277** or **DEKY3277** or **EBKY3277** or **OCKY3277**

3284 Model 1 or 2 Printer for the 3277

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**}
UNIT = **3284**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3284 is attached to a 3274-D control unit. This feature may improve the performance of the 3284 if programs that access the device use the appropriate channel programs.	**PTREAD**

3286 Model 1 or 2 Printer for the 3277

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**}
UNIT = **3286**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3286 is attached to a 3274-D control unit. This feature may improve the performance of the 3286 if programs that access the device use the appropriate channel programs.	**PTREAD**

3287 Model 1, 1C, 2, or 2C Printer

You must define this device as either a 3284 or a 3286. If you plan to use this device as an operator's console, IBM recommends that you define it as a 3286-2 (UNIT=3286,MODEL=2).

You must specify:

- The Model 1C as a Model 1
- The Model 2C as a Model 2

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**}
UNIT = {**3284**|**3286**}

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3287 is attached to a 3274-D control unit. This feature may improve the performance of the 3287 if programs that access the device use the appropriate channel programs.	**PTREAD**

3288 Printer

You must define this device as either a 3284 or a 3286. If you plan to use this device as an operator's console, IBM recommends that you define it as a 3286-2 (UNIT=3286,Model=2).

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**}
UNIT = {**3284**|**3286**}

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3288 is attached to a 3274-D control unit. This feature may improve the performance of the 3288 if programs that access the device use the appropriate channel programs.	**PTREAD**

3289 Model 1 or 2 Printer

You must define this device as either a 3284 or a 3286. If you plan to use this device as an operator's console, IBM recommends that you define it as a 3286-2 (UNIT=3286,MODEL=2).

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**1**|**2**}
UNIT = {**3284**|**3286**}

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Character generator (Select only one)	
United States English (default)	**DOCHAR**
French	**FRCHAR**
German	**GRCHAR**
Katakana	**KACHAR**
United Kingdom	**UKCHAR**
The 3289 is attached to a 3274-D control unit. This feature may improve the performance of the 3289 if programs that access the device use the appropriate channel programs.	**PTREAD**

3290 Information Panel

- You must define the 3290 as a 3270-X. How to do this is explained in "3270 Model X" on page 6-31.
- To customize the 3274 control unit, you must answer a series of questions about the configuration of the devices that are attached to the control unit. For MCS operator consoles, you must answer question 173 so that bit 7 (update panel before allowing buffer change) will be set to one. Having bit 7 set to one ensures that messages previously sent to the 3290 are displayed before new messages are accepted. In other words, setting bit 7 to one prevents the 3290 from losing messages.

5080 Graphics Device

- This topic explains how to define the 5080 so it can be used in full- function mode. The 5080, when defined in full-function mode, is also called a high function graphics device (HFGD).

  The system on which you run the MVS configuration program to define the 5080, and the system on which you use the 5080 must both have Graphics Access Method/SP2 (GAM/SP2) installed.

  For more information about required products, see "Other Products May be Required" on page 6-5.

- You can also define the 5080 so it can be used in 3250-compatibility mode. This use of the 5080 requires that you define it as a 2250-3. For information about defining a 2250-3, see "2250 Model 3" on page 6-26.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **HFGD**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

Magnetic Tape Device Definitions

This topic explains how to code IODEVICE statements for supported magnetic tape devices. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending numerical order by device type. The heading at the top of each page identifies the class of device, magnetic tape, the device type, and model.

3420 Model 3, 5, or 7 Magnetic Tape Unit

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**3**|**5**|**7**}
UNIT = **3420**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-49.)
OFFLINE = {**YES**|**NO**}

Note: If you specify the SHARABLE feature, the MVS configuration program forces the 3420 offline even if you specify OFFLINE = NO.

PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
There is a separate physical control unit path that can be used.	**ALTCTRL**
Data conversion **If you specify DATACONV, you must also specify 7-TRACK.**	**DATACONV**
The tape unit can be used at either 800BPI or 1600BPI.	**DUALDENS**
The 3803 two-channel switch is used for partitioning. Magnetic tape drives can be shared between two processors. Do not allocate or unload a shared tape drive. OFFLINE=YES is forced	**SHARABLE**
Track type (select only one)	
7-track enables the 3420 to process binary coded decimal (BCD) or binary format tapes. These tapes are compatible with the 727, 729, and 730 tape drives.	**7-TRACK**
9-TRACK enables the 3420 to process tapes in 9 track format. The default is 9-TRACK.	**9-TRACK**

3420 Model 4, 6, or 8 Magnetic Tape Unit

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = {**4**|**6**|**8**}
UNIT = **3420**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}

Note: If you specify the SHARABLE feature, the MVS configuration program forces the 3420 offline even if you specify OFFLINE = NO.

PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path that can be used.	**ALTCTRL**
The 3420 can be used at either 1600 BPI or 6250 BPI.	**OPT1600**
The 3803 two-channel switch is used for partitioning. Magnetic tape drives can be shared between two processors. Do not allocate or unload a shared tape drive. OFFLINE = YES is forced	**SHARABLE**
The 3420 has the 9 track feature.	**9-TRACK**

3422 Magnetic Tape Unit

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3422**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}

Note: If you specify the SHARABLE feature, the MVS configuration program forces the 3422 offline even if you specify OFFLINE = NO.

PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path that can be used.	**ALTCTRL**
The 3422 can be used at either 1600 BPI or 6250 BPI.	**OPT1600**
The 3422 two-channel switch is used for partitioning. Magnetic tape drives can be shared between two processors. Do not allocate or unload a shared tape drive. OFFLINE = YES is forced	**SHARABLE**
The 3422 has the 9 track feature.	**9-TRACK**

3430 Magnetic Tape Unit

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3430**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The 3430 can be used at either 1600 BPI or 6250 BPI.	**OPT1600**
The 3430 has the 9 track feature.	**9-TRACK**

3480 Magnetic Tape Unit

You can define the 3480 to operate in one of two modes:

- **3480 full function mode** -- MVS supports all of the 3480 functions. (Code UNIT=3480).
- **3420 code compatibility mode** -- MVS supports the 3480 as though it were a 3420. All of the 3420 channel programs will execute on the 3480. (Code UNIT=3420C).

Note: *Do not* define both modes within the same I/O configuration.

Note: If you install MVS/Data Facility Product Version 3, Release 2.0, you cannot define the 3480 to operate in the 3420 mode.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = {**3420C**|**3480**}

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}

Note: If you specify the SHARABLE feature, the MVS configuration program forces the 3480 offline even if you specify OFFLINE=NO.

PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
There is a separate physical control unit path that can be used.	**ALTCTRL**
Magnetic tape drives can be shared between two processors. Do not allocate or unload a shared tape drive. OFFLINE=YES is forced	**SHARABLE**
A 3480 can be defined as a 3480X, which can write compacted data.	**COMPACT**

Miscellaneous I/O Device Definitions

This topic explains how to code IODEVICE statements for:

- Supported miscellaneous devices.
- Unsupported devices that you want MVS to treat as DUMMY devices.

The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending numerical order by device type. The heading at the top of each page identifies the class of device, miscellaneous devices, the device type, and model.

DUMMY Device

This book also refers to a DUMMY device as an *unsupported device.*

MVS supports each DUMMY device as a unit record device. For additional information about defining a DUMMY device, see "Defining an Unsupported Device" on page 6-4.

For information about migrating a DUMMY device, see "Migrating a Dummy Device" on page 8-2.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **DUMMY**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

1287 Optical Reader

The system on which you run the MVS configuration program, and the system on which you use the 1287 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **1287**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

1288 Optical Page Reader

The system on which you run the MVS configuration program, and the system on which you use the 1288 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **1288**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3540 Diskette I/O Unit

The system on which you run the MVS configuration program, and the system on which you use the 3540 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3540**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3838 Array Processor

The system on which you run the MVS configuration program, and the system on which you use the 3838 must both have the Vector Processing Subsystem/Extended Architecture (VPSS/XA) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = (*device-number***,1)**

Note: The low-order digit (the rightmost digit) of the *device-number* must be 0. You could code, for example, ADDRESS = (110,1) or ADDRESS = (540,1).

The MVS configuration program assigns eight device numbers to the 3838: the device number you specify plus the next seven consecutive device numbers. For example, if you specify ADDRESS = (210,1), the MVS configuration program assigns the 3838 device numbers 210, 211, 212, 213, 214, 215, 216, and 217.

CUNUMBR = (*number*[,*number*]...**)**
UNIT = 3838

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3848 Cryptographic Unit

The system on which you run the MVS configuration program, and the system on which you use the 3848 must both have the cryptographic unit support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = **1**
UNIT = **3848**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3886 Optical Character Reader

The system on which you run the MVS configuration program, and the system on which you use the 3886 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3886**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3890 Document Processor

The system on which you run the MVS configuration program, and the system on which you use the 3890 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3890**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3895 Document Reader/Inscriber

The system on which you run the MVS configuration program, and the system on which you use the 3895 must both have the magnetic ink character reader/optical character reader (MICR/OCR) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3895**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

MSS Device Definitions

This topic explains how to code IODEVICE statements for the Mass Storage Subsystem. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending numerical order by device type. The heading at the top of each page identifies the class of device, MSS, the device type, and model.

To define the IBM Mass Storage Subsystem, you must define both a IBM 3330 Disk Storage Unit and a IBM 3851 Mass Storage Facility.

3330 Disk Storage Unit (Used as a Virtual Volume in an MSS)

The 3330V is functionally equivalent to the 3330/3333 model 1.

When you define the 3330V you must also define the 3851. For information about defining the 3851, see "3851 Mass Storage Facility" on page 6-66.

The system on which you run the MVS configuration program, and the system on which you use the 3330V must both have the Mass Storage Subsystem/Extended Architecture (MSS/XA) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3330V**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
There is a separate physical control unit path to the device	**ALTCTRL**
DASD sharing and reserve/release logic usage: (Specify only one.)	
Share device with other systems; Use reserve/release logic as needed.	**SHARED**
Use reserve/release logic only when the system is physically partitioned.	**SHAREDUP**

3851 Mass Storage Facility

To fully define the 3851 mass storage facility:

- Code an IODEVICE statement to define the 3851 as explained below.
- Code an IODEVICE statement to define the 3330V as explained in the topic "3330 Disk Storage Unit (Used as a Virtual Volume in an MSS)" on page 6-65.
- During the SYSGEN process for the system on which the 3851 will be used, specify MSS=YES on the DATAMGT macro instruction. For more information on the DATAMGT macro instruction, see *System Generation*

The system on which you run the MVS configuration program, and the system on which you use the 3851 must both have the Mass Storage Subsystem/Extended Architecture (MSS/XA) support installed.

For more information about required products, see "Other Products May be Required" on page 6-5.

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3851**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

Telecommunication Device Definitions

This topic explains how to code IODEVICE statements for supported telecommunication devices. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending alphanumerical order by device type. The heading at the top of each page identifies the class of device, telecommunication, the device type, and model.

> **Note:** Before defining any telecommunication devices, read the following topic "Defining Telecommunication Devices." That topic provides information about those devices that you must define through the MVS configuration program and those that you must not.

Defining Telecommunication Devices

All telecommunication devices attached to a given line must be of the same type. All of these devices must also have the same features.

To define telecommunication devices, code one IODEVICE statement for each:

- Telecommunications line. *Do not* code an IODEVICE statement for each device on the line. One IODEVICE statement is all that is needed to define one line and all of the devices on that line. Use the UNIT parameter on the IODEVICE statement to specify the type of device that is on the line.
- Locally attached IBM 3704 Communications Controller
- Locally attached IBM 3705 Communications Controller
- Locally attached IBM 3791 Communications Controller

3704, 3705 Emulation Program Considerations

If you plan to use a locally attached 3704 or 3705 with the emulation program, you must code one IODEVICE statement for the 3704 or 3705. You must also code an IODEVICE statement for each line that will operate in emulation mode.

Virtual Telecommunications Access Method (VTAM) Considerations

In a VTAM telecommunications network, you need code IODEVICE statements only for:

- Each locally attached IBM 3270 Information Display System
- Each locally attached IBM 3704 or 3705 Communication Controller

Other devices that are included in the network are not defined through the MVS configuration program. Instead, they are defined through the VTAM network definition process.

Note: VTAM network definition is a separate process. For more information, see the VTAM and Network Control Program (NCP) books that apply to the version and release level of the VTAM and NCP products that are installed.

> **Note:** There is one exception to the previously stated rule about locally attached 3704 or 3705 Communications Controllers in a VTAM network: if a locally attached 3704 or 3705 in a VTAM network also supports a non-VTAM network through an NCP with the partitioned emulation programming (PEP) extension, you must code an IODEVICE statement for each line that will be operated in emulation mode.

BSC1, BSC2, and BSC3

Required Parameters

ADAPTER = BSCA
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Notes:

1. If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703. (Specify 2701 for JES2 BSC RJE lines.)
2. For BSC1 only: if you specify either TCU = 2702 or TCU = 2703, omit the FEATURE parameter.

UNIT = {**BSC1**|**BSC2**|**BSC3**}

Notes:

1. For devices connected to a *nonswitched point-to-point line*, code BSC1.
2. For devices connected to a *switched point-to-point line*, code BSC2.
3. For devices connected to a *nonswitched multipoint line*, code BSC3.

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-70.)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered. **This is the default for BSC2.** **If you specify UNIT=BSC3, do not specify AUTOANSR.**	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit. **Code for BSC2 only.**	**AUTOCALL**
The TCU has an automatic polling feature. **This is the default for BSC3.** **If you specify UNIT=BSC2, do not specify AUTOPOLL.**	**AUTOPOLL**
The TCU has the dual code feature. **Specify DUALCODE only if the device is connected to a 2701.**	**DUALCODE**
The TCU has the dual communication interface feature. **Specify DUALCOMM only if the device is connected to a 2701.**	**DUALCOMM**

TWX (Teletype) Model 33 or 35

Required Parameters

ADAPTER = TELE2
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = TWX

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**

[1] Trademark of Teletype Corporation.

WTTA (IBM World Trade Corporation Telegraph Terminal)

Required Parameters

ADAPTER = TELEW
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = WTTA

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

1030 Data Collection System

Required Parameters

ADAPTER = IBM2
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 1030

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The TCU has an automatic polling feature. Specify this feature only for lines connected through a type II synchronous data adapter.	**AUTOPOLL**

1050 Data Communications System

Required Parameters

ADAPTER = {**IBM1**|**IBMT**}

Notes:

1. If the connection is through a terminal adapter or a line adapter, code IBM1.
2. If the connection is through a telegraph adapter or a telegraph line adapter, code IBMT.

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Notes:

1. If you code ADAPTER = IBMT, code either TCU = 2701 or TCU = 2703.
2. If you code ADAPTER = IBM1 and the telecommunications line connects to a 3705 that is operating in emulator mode, code either TCU = 2702 or TCU = 2703.

UNIT = **1050**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-75.)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**
The TCU has an automatic polling feature. Specify this feature only for lines connected through a type II synchronous data adapter. **If you specify either AUTOANSR, AUTOCALL, or both,** *do not* **specify AUTOPOLL**	**AUTOPOLL**

1050X (1050 Data Communications System with Timeout Suppression)

Required Parameters

ADAPTER = {**IBM1**|**IBMT**}

Notes:

1. If the connection is through a terminal adapter or a line adapter, code IBM1.
2. If the connection is through a telegraph adapter or a telegraph line adapter, code IBMT.

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Notes:

1. If you code ADAPTER = IBMT, code either TCU = 2701 or TCU = 2703.
2. If you code ADAPTER = IBM1 and the telecommunications line connects to a 3705 that is operating in emulator mode, code either TCU = 2702 or TCU = 2703.

UNIT = **1050X**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**

115A (Western Union Terminal)

Required Parameters

ADAPTER = TELE1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 115A

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

2 Trademark of Western Union Telegraph Company.

2740 Communication Terminal

Required Parameters

ADAPTER = IBM1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 2740

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features" on page 6-79.)
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**
The TCU has an automatic polling feature. Specify this feature only for lines connected through a type II synchronous data adapter. **If you specify AUTOPOLL, also specify SCONTROL.** If you specify either AUTOANSR, **AUTOCALL**, or both, do not specify AUTOPOLL.	**AUTOPOLL**
The terminal has the record checking feature. **If you specify OIU, you must also specify CHECKING.**	**CHECKING**
The terminal is a model 1 with RPQ #S30031 installed.	**INTERRUPT**
The terminal is equipped with the IBM 2760 optical image unit. **You must also specify CHECKING. Besides CHECKING, the only other features you can specify are AUTOANSR and AUTOCALL.**	**OIU**
The terminal is equipped with the station control feature. **If you specify SCONTROL,** *do not* **specify AUTOANSR, AUTOCALL, OIU, or XCONTROL.**	**SCONTROL**
The terminal is equipped with the dial-up feature. **You must also specify AUTOANSR, AUTOCALL, or both as appropriate.** *Do not* **specify OIU.**	**XCONTROL**

2740C (2740 Communication Terminal with Correspondence Code)

Required Parameters

ADAPTER = IBM1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
FEATURE = CHECKING|(CHECKING[,*feature*]...) (See "Device Features.")
TCU = {**2701|2702|2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 2740C

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0|1|2|3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**
The terminal has the record checking feature. **CHECKING is required**	**CHECKING**

2740X (2740 Communication Terminal with PTTC Code)

Required Parameters

ADAPTER = IBM1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...**)**
FEATURE = CHECKING|(CHECKING[,*feature*]...**)** (See "Device Features.")
TCU = {**2701|2702|2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 2740X

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**
The TCU has the autocall feature and the line is connected to the TCU terminal adapter through an automatic calling unit.	**AUTOCALL**
The terminal has the record checking feature. **CHECKING is required**	**CHECKING**

3 Perforated tape and transmission code.

2741C (2741 Communication Terminal with Correspondence Code)

Required Parameters

ADAPTER = IBM1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 2741C

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**

2741P (2741 Communication Terminal with PTTC Code)

Required Parameters

ADAPTER = IBM1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 2741P

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The modem that connects the TCU to the telecommunication line is a switched line over which calls are to be answered.	**AUTOANSR**

3274 Model 1A Control Unit

1. If the 3274-1A is locally attached and it has systems network architecture (SNA) support that permits the control of multiple devices through one device number, code an IODEVICE statement for the 3274-1A.

   You must define the 3274-1A as though it were a 3791L. For instructions on how to code the IODEVICE statement, see "3791L Controller" on page 6-89.

   Do not code IODEVICE statements for any of the devices attached to the 3274-1A.

2. If each device that is attached to the 3274-1A has its own device number, do not code an IODEVICE statement for the 3274-1A. You must, instead, code an IODEVICE statement for each of the attached devices. For instructions on how to code the IODEVICE statements, see the appropriate device descriptions in this chapter.

3704 Communications Controller

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3704**

Optional Parameters

ADAPTER = **CA1**
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3705 Communications Controller

Required Parameters

ADAPTER = {**CA1**|**CA2**}

Notes:

1. If the channel adapter is a type 3, code **CA2.**
2. If the channel adapter is a type 4, code **CA1.**

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3705**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3725 Communications Controller

The 3725 is functionally equivalent to a 3705. Therefore, you must define the 3725 just as though it were a 3705. For instructions on how to code the IODEVICE statement, see "3705 Communications Controller" on page 6-86.

3767 Communication Terminal

The 3767 model 1 or 2 is functionally equivalent to a 2740 model 1. You must define the 3767 just as though it were a 2740 model 1. For instructions on how to code the IODEVICE statement, see "2740 Communication Terminal" on page 6-78.

3791L Controller

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3791L**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

7770 Audio Response Unit

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
MODEL = **3**
UNIT = **7770**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

83B3 (AT&T Selective Calling Terminal)

Required Parameters

ADAPTER = TELE1
ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
TCU = {**2701**|**2702**|**2703**}

Note: If the telecommunications line connects to a 3705 that is operating in emulator mode, specify 2702 or 2703.

UNIT = 83B3

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
SETADDR = {**0**|**1**|**2**|**3**}

Note: If you code TCU = 2702, also code the SETADDR parameter.

STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

4 Trademark of American Telephone and Telegraph Company

Unit Record Device Definitions

This topic explains how to code IODEVICE statements for supported unit record devices. The documentation for each device starts on a new page and includes:

- Required parameters
- Optional parameters
- Device features (if applicable)

Devices are arranged in ascending alphanumerical order by device type. The heading at the top of each page identifies the class of device, the device type, and model.

AFP1 Model 0

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
MODEL = **0**
UNIT = **AFP1**

Note: The AFP1 unit type is used for: 3825, 3827, and 3835.

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

1403 Model N1, 2, or 7 Printer

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
MODEL = {**N1**|**2**|**7**}
UNIT = **1403**

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Universal character set **Note:** Do not specify this feature for the Model 7.	**UNVCHSET**

2501 Model B1 or B2 Card Reader

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
MODEL = {**B1**|**B2**}
UNIT = **2501**

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {YES|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|N}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Read in card image mode	**CARDIMAGE**

2540 Model 1 Card Read Punch

To define the 2540, code two **IODEVICE** statements:

- One to define the reader
- One to define the punch

Specify the **UNIT=** parameter as follows:

- For the reader specify **UNIT=2540R**.
- For the punch specify **UNIT=2540P**.

Required Parameters

ADDRESS={*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR=(*number*[,*number*]....)
UNIT={**2540R**|**2540P**}

Optional Parameters

FEATURE=*feature* (See "Device Features.")

MODEL=**1**
OFFLINE={**YES**|**NO**}
PATH=*chpid*
STADET={**Y**|**N**}
TIMEOUT={**Y**|**N**}
UNITADD=*nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE=
Read or punch in card image mode **Note:** Specify only if the 2821 control unit has the column-binary feature.	**CARDIMAGE**

3203 Model 5 Printer

You can define this device as a model 4 or a model 5.

Required Parameters

ADDRESS = {*device-number*|(*device-number*,*number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
MODEL = {**4**|**5**}
UNIT = **3203**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3211 Printer

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **3211**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3262 Model 5

MVS/XA supports the 3262-5 as a 4248 printer.

Required Parameters

ADDRESS. = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **4248**

Note: You must define the 3262 model 5 as a 4248 (no model number).

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

3505 Card Reader

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **3505**

Optional Parameters

FEATURE = *feature* (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Read in card image mode	**CARDIMAGE**

3525 Card Punch

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]...)
UNIT = **3525**

Optional Parameters

FEATURE = (*feature*[,*feature*]...) (See "Device Features.")
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
Read in card image mode	**CARDIMAGE**
Print feature (Select only one)	
print up to 2 lines per card	**TWOLINE**
print up to 25 lines per card	**MULTILINE**

3800 Model 1, 3, 6, or 8 Printing Subsystem

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **3800**

Optional Parameters

FEATURE = (*feature*[,*feature*]....) (See "Device Features.")

MODEL = {**1**|**3**|**6**|**8**}
OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

To indicate the presence of this feature:	code this value on the parameter, FEATURE =
The burster-trimmer-stacker feature is installed.	**BURSTER**
Character generation storage **(Select only one)** Do not specify this feature for models 3 or 8.	
1-character	**CGS1**
2-character	**CGS2**

4245 Printer

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **4245**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

4248 Printer

Required Parameters

ADDRESS = {*device-number*|(*device-number,number-of-units*)}

Note: The default for *number-of-units* is 1.

CUNUMBR = (*number*[,*number*]....)
UNIT = **4248**

Optional Parameters

OFFLINE = {**YES**|**NO**}
PATH = *chpid*
STADET = {**Y**|**N**}
TIMEOUT = {**Y**|**N**}
UNITADD = *nnn*

Device Features

None supported.

Chapter 7. Defining Eligible Device Tables and NIP Consoles

Each I/O configuration can define one or more eligible device tables (EDT) and must define at least one device that NIP will be allowed to use as a console.

For each I/O configuration, you can define:

- One EDT with no changes to the device preference table
- One EDT with changes to the device preference table
- Multiple EDTs with or without changes to the device preference tables
- One or more NIP consoles

Each EDT definition can include:

- Definitions for one or more esoteric device groups
- Changes to the device preference table
- Names of the generic device types you want to make eligible for VIO

Locating Information in this Chapter

This chapter contains the following major topics:

Defining One EDT with No Device Preference Table Changes

To define this EDT, you may either include an EDT statement in the input stream or omit it. If you want the EDT identifier to default to the I/O configuration identifier (specified on the IOCONFIG statement), omit the EDT statement.

To assign the EDT a specific identifier, code the identifier on the ID= parameter of an EDT statement. Place the EDT statement in the input stream immediately after the last IODEVICE statement.

Examples

The following input streams illustrate both ways to define this EDT.

- This example shows an input stream with no EDT statement. The MVS configuration program assigns the EDT an identifier of 01, the same as the I/O configuration identifier.

```
IOCONFIG ID=01
ID...
CHPID...
CNTLUNIT...
IODEVICE...
   .
   .
IODEVICE...
UNITNAME...
   .
   .
UNITNAME...
NIPCON...
```

- This example shows the same input stream but this time it includes an EDT statement. This EDT will have an identifier of 02.

```
IOCONFIG ID=01
ID...
CHPID...
CNTLUNIT...
IODEVICE...
   .
   .
IODEVICE...
EDT ID=02
UNITNAME...
   .
   .
UNITNAME...
NIPCON...
```

Defining One EDT with Device Preference Table Changes

To define this EDT, include an EDT statement in the input stream. If you want the EDT identifier to default to the I/O configuration identifier (specified on the IOCONFIG statement), omit the ID= parameter from the EDT statement. To assign a specific identifier, specify the identifier on the ID= parameter of the EDT statement.

To change the device order that MVS is to use when attempting to satisfy an allocation request, use the DEVPREF= parameter on the EDT statement. When MVS attempts an allocation, it attempts to allocate in the order you have specified on the DEVPREF= parameter. If the allocation is unsuccessful, MVS continues the attempt following the order specified in the IBM-provided device preference table. For a description of the IBM-provided device preference table, see Appendix B, "IBM Provided Device Preference Table" on page B-1.

Place the EDT statement in the input stream immediately after the last IODEVICE statement.

Example

The following example defines one EDT and changes the device preference table. The EDT identifier defaults to the identifier of the I/O configuration, A0.

```
IOCONFIG ID=A0
ID....
CHPID....
CNTLUNIT....
IODEVICE....
   .
   .
IODEVICE....
EDT DEVPREF=(3375,3380)
UNITNAME....
   .
   .
UNITNAME....
NIPCON....
```

Defining Multiple EDTs

> **Note:** JES3 supports only one EDT per I/O configuration. If a JES3 installation defines more than one EDT in a configuration, JES3 will support only the first EDT defined.

To define multiple EDTs, include an EDT statement for each EDT. Each EDT must have an identifier that is different than all other EDT identifiers. In other words, no two EDTs are allowed to have the same identifier.

To assign the EDT identifier, use the ID= parameter on the EDT statement. You may omit the ID= parameter from one (but no more) of the EDT statements. If you omit the ID= parameter, the MVS configuration program assigns the EDT the same identifier as the I/O configuration (specified by ID= on the IOCONFIG statement).

To change the device order that MVS is to use when attempting to satisfy an allocation request, use the DEVPREF= parameter on the EDT statement. If the allocation is unsuccessful, MVS continues the attempt following the order specified in the IBM-provided device preference table. For a description of the IBM-provided device preference table, see Appendix B, "IBM Provided Device Preference Table" on page B-1.

The topic "Grouping EDT and UNITNAME Statements" on page 7-8 explains how to group EDT statements with UNITNAME statements to define multiple EDTs.

Example

This example defines three EDTs.

```
IOCONFIG ID=B1
ID....
CHPID....
CNTLUNIT....
IODEVICE....
   .
   .
IODEVICE....
EDT DEVPREF=(4245,1403,3375)
UNITNAME....
  .
UNITNAME....
EDT ID=01,DEVPREF=(3330,3380)
UNITNAME....
  .
UNITNAME....
EDT ID=02
UNITNAME....
   .
UNITNAME....
NIPCON....
```

The first EDT statement omits the ID= parameter and changes the device preference table. The identifier of the corresponding EDT will default to B1.

The second EDT statement assigns the corresponding EDT an identifier of 01 and it also changes the device preference table.

The third EDT statement assigns an identifier of 02 but does not change the device preference table.

Defining Esoteric Device Groups

When you define an esoteric device group, you name it and identify the devices that the group is to include. You can also make the group eligible for VIO.

To define an esoteric device group, use the UNITNAME statement. The NAME= parameter names the group. The UNIT= parameter specifies the device numbers of the devices you want included in the group. If the group contains at least one DASD and you want to make the group eligible for VIO, code VIO=YES.

All of the devices that you assign to an esoteric device group must be of the same device class with the following exception: you may define esoteric device groups that contain devices from both the DASD and magnetic tape device classes. All devices that you assign must also be defined on IODEVICE statements.

After you have defined all of the esoteric device groups for a particular EDT, place the UNITNAME statements immediately after the corresponding EDT statement. Repeat this procedure, defining esoteric device groups and placing the UNITNAME statements immediately after the corresponding EDT statement, until you have defined all of the esoteric groups you wish to define.

Required Esoteric Device Groups

A number of procedures in the IBM-provided SYS1.PROCLIB refer to the esoteric names SYSDA or SYSSQ. If your installation uses these procedures, you must define these esoteric device groups to the MVS configuration program.

The group named SYSSQ must include only magnetic tapes, only direct access devices, or both. The group named SYSDA must include direct access devices only.

In addition, your installation might use other cataloged procedures that also refer to these or to other esoteric names.

To determine which esoteric groups your installation uses, review the UNIT= parameter on the cataloged procedure DD statements.

Naming Esoteric Device Groups

Within a given EDT, no two esoteric device groups can have the same name. An esoteric device group in one EDT , however, can have the same name as an esoteric device group in another EDT.

Example 1

This example contains two esoteric device group definitions. One group, GROUPA, is ineligible for VIO (that is the default when you omit the VIO= parameter). The other group, GROUPB, is eligible for VIO (VIO=YES).

```
      .
      .
      .
UNITNAME NAME=GROUPA,UNIT=(210,211)
      .
UNITNAME NAME=GROUPB,UNIT=(312,414,416),VIO=YES
      .
```

Example 2

This example illustrates the naming convention that states "An esoteric device group in one EDT can have the same name as an esoteric device group in another EDT". In this example, both EDTs define esoteric device groups named **SYSSQ**, **SYSDA**, and **GROUPA**.

```
IOCONFIG ID=01
   .
   .
   .
   .
EDT
UNITNAME NAME=SYSDA,...
UNITNAME NAME=SYSSQ,...
UNITNAME NAME=GROUPA,...
UNITNAME NAME=GROUPB,...
EDT ID=02
UNITNAME NAME=SYSDA,...
UNITNAME NAME=SYSSQ,...
UNITNAME NAME=GROUPA,...
NIPCON...
```

Identifying VIO-Eligible Generic Device Types

You can identify one or more *direct access* generic device types that you want to make eligible for VIO. You define each generic device type on a UNITNAME statement.

- Specify the generic name on the NAME= parameter. For a list of IBM-defined generic names, see Appendix A, "Generic Device Types" on page A-1.
- Always code VIO=YES.
- *Do not* code the UNIT= parameter.

Place these UNITNAME statements in the input stream with other UNITNAME statements following the corresponding EDT statement. If the input stream contains no EDT statements, place the UNITNAME statements with other UNITNAME statements.

The following example shows how to code the UNITNAME statement to make the generic device type 3350 eligible for VIO.

```
UNITNAME NAME=3350,VIO=YES
```

Grouping EDT and UNITNAME Statements

The following example defines two EDTs and shows how to group corresponding **UNITNAME** statements.

```
         IOCONFIG ID=A0
*
         ID....
         CHPID....
         CNTLUNIT....
*
         IODEVICE....
             .
             .
             .
         IODEVICE....
*
         EDT ID=01
*
* THE FOLLOWING UNITNAME STATEMENTS DEFINE THE ESOTERIC
* DEVICE GROUPS AND VIO-ELIGIBLE GENERIC DEVICE TYPES
* FOR EDT 01
*
         UNITNAME....
         UNITNAME....
         UNITNAME....
*
         EDT ID=02
*
* THE FOLLOWING UNITNAME STATEMENTS DEFINE THE ESOTERIC
* DEVICE GROUPS AND VIO-ELIGIBLE GENERIC DEVICE TYPES
* FOR EDT 02
*
         UNITNAME....
         UNITNAME....
         UNITNAME....
*
         NIPCON
```

Defining NIP Consoles

For each I/O configuration, you must define one or more devices that NIP will be allowed to use as a console. Figure 7-1 lists the device types and models that you can define as NIP consoles.

Note: To ensure that all system messages appear on one console, define the master console as the first available NIP console. You use the MVS Configuration Program to define only those consoles that you will allow NIP to use. For information about how to define the master console, see *Initialization and Tuning*.

Figure 7-1. Devices That Can be Defined as NIP Consoles

Device and model	Description
2740-1	Communications terminal - The terminal *must* have the CHECKING and OIU features and you *must* specify these features.
3270-X	Any device that can be defined as a 3270-X. NIP uses this device as though it were a 3277-2.
3277-2	Display station
3278-2, -2A, -3, -4, or -5	Display station
3279-2A, -2B, -2C, -3A, or -3B	Display station

To define a device as a NIP console, code the device number of the device on a NIPCON statement. NIP attempts to use the devices in the order they appear on the NIPCON statement. If the list of device numbers will not fit on one NIPCON statement, use multiple statements. The NIPCON statement is described in "NIPCON Statement" on page 5-15.

Each device number that appears on a NIPCON statement must also be defined on an IODEVICE statement.

Nipcon example: The following example shows how to code the NIPCON statement. NIP attempts to use the devices in the order shown, 100, 210, 202, and 250. Because device number 100 is the first NIP console defined, it should also be defined as the master console.

```
NIPCON DEVNUM=(100,210,202)

NIPCON DEVNUM=250
```

Chapter 8. I/O Configuration Management Tasks

The tasks in this chapter support the procedures described in Chapter 4, "Managing I/O Configurations" on page 4-1. You will do one or more of these tasks each time you use those procedures.

Locating Information in this Chapter

The chapter contains the following major topics:

Using SYSGEN Input Statements to Migrate a Configuration

There are several input statements that you previously used with the SYSGEN process that you use as input to the MVS configuration program:

- **CHPID**
- **CNTLUNIT**
- **ID**
- **IODEVICE**
- **UNITNAME**

The MVS configuration program accepts the CHPID, CNTLUNIT, ID, and UNITNAME statements without change.

To determine if changes are required to the IODEVICE statements, read the rest of this topic.

Migrating a Dummy Device

It is possible that the SYSGEN input stream will contain IODEVICE statements that specify UNIT=DUMMY. If you use these statements and continue to allow the UNIT= parameter to specify UNIT=DUMMY, the MVS configuration program will define the device as though it were a unit record device.

Note: When you used the SYSGEN process to define a DUMMY device, you could code the DEVTYPE parameter and the ERRTAB parameter on the IODEVICE statement. DEVTYPE defined the UCBTYP field; ERRTAB identified the error routine to be used for the device. The MVS configuration program ignores both of the parameters.

The MVS configuration program assigns all DUMMY devices:

- A UCBTYP value of X'00000800'.
- An error routine value of X'0D'. The MVS configuration program converts this value to an error recovery program (ERP) name of IGE0001C.
- A generic name of DUMMY.

To define the device as something other than a DUMMY device, you must:

- Provide a program called a unit information module (UIM)
- Recode the IODEVICE statement according to the design of the UIM

The UIM design defines the parameters that you can code on the IODEVICE statement including the value that you code for UNIT=.

For information about coding an IODEVICE statement, see "IODEVICE Statement" on page 5-9. For information about coding a UIM and adding it to your system, see *System Modifications*.

Removing Unsupported Parameters

Possibly, some of the IODEVICE statements that you use from the SYSGEN input stream will contain parameters that MVS no longer supports. The unsupported parameters are **AP, DEVTYPE, ERRTAB, EXPBFR, OBRCNT, and OPTCHAN**. If you allow these parameters to remain, the MVS configuration program ignores the parameters and issues informational messages. Otherwise, the MVS configuration program works normally.

To avoid receiving these informational messages, remove any unsupported parameters.

Assigning an I/O Configuration Identifier

When you migrate an I/O configuration or define a new I/O configuration, you must assign an identifier to the new configuration that the MVS configuration program builds. The identifier is a two-character suffix that the MVS configuration program appends to the I/O configuration member names: IEANCTxx, IOSIITxx, and IOSUCBxx (xx represents the appended suffix). This identifier enables installation personnel and MVS to distinguish one configuration from another.

To prevent the MVS configuration program from writing over an existing I/O configuration, always assign an identifier that is unused. If you assign an identifier that's currently in use, the MVS configuration program replaces the existing configuration with the new configuration, thus destroying the existing configuration.

To assign the identifier, use the **ID=** parameter on the **IOCONFIG** statement. The **IOCONFIG** statement is described in "IOCONFIG Statement" on page 5-7.

IOCONFIG Example: This example assigns an identifier of 00 to a configuration.

```
IOCONFIG ID=00
```

Notes:

1. If the input stream contains no EDT statement, the MVS configuration program assigns the EDT the same identifier as the I/O configuration.
2. If the input stream contains an EDT statement from which you have omitted the ID= parameter, the MVS configuration program assigns the corresponding EDT the same identifier as the I/O configuration.

Using the Cataloged Procedure or the SAMPLIB JCL

To help you run the MVS configuration program, IBM provides a cataloged procedure in a SYS1.PROCLIB member and JCL statements in two SYS1.SAMPLIB members. The decision whether to use the cataloged procedure or the JCL depends on two factors:

- The release level of the system on which you run the MVS configuration program.
- The release level of the system on which the I/O configuration will be used.

The following table will help you decide whether to use the cataloged procedure or the SAMPLIB JCL:

What You Are Doing	What to Use
If you intend to run the MVS configuration program on an MVS/XA or MVS/ESA release level that is the same as the release level on which the I/O configuration will be used: (for example, you intend to run the MVS configuration program on MVS/SP3.1.0 to build a configuration for MVS/SP3.1.0),	Use SYS1.PROCLIB(MVSCP). See "SYS1.PROCLIB(MVSCP) Cataloged Procedure" on page 8-5.
If you intend to run the MVS configuration program on an earlier MVS/XA release level than the release level on which the I/O configuration will be used: (for example, you intend to run the MVS configuration program on MVS/SP2.2.0 to build a configuration for MVS/SP3.1.0),	Use SYS1.SAMPLIB(MVSCPXA). See "SYS1.SAMPLIB(MVSCPXA) JCL Statements" on page 8-7.
If you intend to run the MVS configuration program on MVS/370,	Use SYS1.SAMPLIB(MVSCP370). See "SYS1.SAMPLIB(MVSCP370) JCL Statements" on page 8-10.

Note: One or more macros that the MVS configuration program uses may have been previously updated by a program temporary fix (PTF). If this is the case, and your installation has applied but not yet accepted the PTF, concatenate SYS1.SMPMTS ahead of SYS1.MACLIB in the cataloged procedure or sample JCL, whichever you use. This concatenation ensures that MVSCP uses the updated version of each macro.

SYS1.PROCLIB(MVSCP) Cataloged Procedure

The IBM-provided cataloged procedure contains the following JCL statements:

```
//MVSCP    PROC  TYPE='VALIDATE',DEFDATA='',JES3='NULLFILE'
//MVSCP    EXEC  PGM=CBPMVSCP,PARM='&TYPE'
//SYSPRINT DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=80,BLKSIZE=1600)
//SYSIN    DD    DSN=&DEFDATA,DISP=SHR
//SYSLMOD  DD    DSN=SYS1.NUCLEUS,DISP=SHR
//SYSLIB   DD    DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD    UNIT=SYSDA,SPACE=(TRK,(15,10))
//SYSUT2   DD    UNIT=SYSDA,SPACE=(TRK,(15,10)),DCB=(RECFM=FB,LRECL=80,
//                BLKSIZE=1680)
//SYSUT3   DD    UNIT=SYSDA,SPACE=(TRK,(30,10))
//SYSASMP  DD    SYSOUT=*
//SYSLOUT  DD    SYSOUT=*
//SYSLKEDP DD    SYSOUT=*
//SYSRPT   DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=2926)
//SYSJES3  DD    DSN=&JES3,DISP=SHR
```

Figure 8-1. Cataloged Procedure JCL in SYS1.PROCLIB(MVSCP)

The PROC statement contains three parameters:

- **TYPE=** specifies the mode of operation. The default is VALIDATE. For information about the purpose of each mode, see "Selecting the Mode of Operation" on page 8-15.
- **DEFDATA=** specifies the name of the input data set. That is the data set that contains the input statements. You *must* always specify the data set name.
- **JES3=** specifies the name of the data set where the MVS configuration program is to write the data for the JES3 initialization stream checker. Specify this name only if you want this output written. The default is NULLFILE (no output is written).

To accommodate devices with different output widths, override the LRECL parameter on the SYSPRINT DD statement. Valid LRECL values are in the range 80 to 255. The LRECL value determines the format of MVS configuration program messages.

BLKSIZE be an even multiple of LRECL. Therefore, if you change LRECL, you may also have to change BLKSIZE.

The following table explains the use of the data sets that the cataloged procedure allocates.

Figure 8-2. How the MVS Configuration Program Uses the Data Sets

DD Name	How the Data Set is Used
SYSPRINT	Contains messages issued by the MVS configuration program.
SYSIN	Contains the MVS configuration program input statements.
SYSLMOD	Contains the I/O configuration built by the MVS configuration program. (This is the linkage editor SYSLMOD file).
SYSLIB	Contains the MVS configuration program parsing macros. (This is the assembler SYSLIB file).
SYSUT1	Used by the assembler and the linkage editor as a work file.
SYSUT2	The MVS configuration program, before processing the input stream, writes the input stream to this file. The file then becomes the assembler SYSIN file.
SYSUT3	This file serves several purposes: • It is the assembler SYSPUNCH file: the assembler writes its output to this file which then becomes the loader SYSLIN file. • It is the linkage editor SYSLIN file: the MVS configuration program writes the I/O configuration object modules to this file which then becomes the linkage editor SYSLIN file.
SYSASMP	Contains the listing written by the assembler. (This is the assembler SYSPRINT file.)
SYSLOUT	Contains messages issued by the loader. (This is the loader SYSLOUT file.) The loader writes this file only when the MVS configuration program runs in DEBUG mode.
SYSLKEDP	Contains linkage editor diagnostic output. (This is the linkage editor SYSPRINT file.) The linkage editor writes this file only when the MVS configuration program runs in IODEF mode or EDT mode.
SYSRPT	Contains the MVS configuration program reports: • The IOS report. • The allocation report. • The NIP report.
SYSJES3	Contains data, written by the MVS configuration program, that will be read by the JES3 initialization stream checker.

SYS1.SAMPLIB(MVSCPXA) JCL Statements

To run the MVS configuration program on an earlier MVS/XA release level than the release level on which the I/O configuration will be used, use the SYS1.SAMPLIB(MVSCPXA) JCL. The SAMPLIB JCL contains the same DDs as the cataloged procedure (the use of the data sets allocated by those DDs is explained in Figure 8-2 on page 8-5) plus a STEPLIB DD that defines the SYS1.LINKLIB data set. The SYS1.LINKLIB data set must contain MVS configuration program modules that are at the same release level as the system on which the I/O configuration will be used.

Before using the SAMPLIB JCL, however, you must modify some of the JCL statements. For instructions on how to modify those statements, see "Modifying SYS1.SAMPLIB(MVSCPXA) JCL" on page 8-9.

```
//*********************************************************************
//*                                                                   *
//*  THIS JCL SHOULD BE USED TO RUN THE MVSCP ON AN EARLIER MVS/XA    *
//*  RELEASE LEVEL THAN THE LEVEL ON WHICH THE I/O CONFIGURATION      *
//*  WILL BE USED.                                                    *
//*                                                                   *
//*  IF YOU ARE RUNNING THE MVSCP ON AN MVS/XA RELEASE LEVEL THAT     *
//*  IS THE SAME AS THE RELEASE LEVEL ON WHICH THE I/O CONFIGURATION  *
//*  WILL BE USED, THEN USE THE CATALOGED PROCEDURE THAT IS IN        *
//*  SYS1.PROCLIB(MVSCP).                                             *
//*                                                                   *
//*  IF YOU ARE RUNNING THE MVSCP ON AN MVS/370 SYSTEM, THEN USE THE  *
//*  SAMPLE JCL IN SYS1.SAMPLIB(MVSCP370).                            *
//*                                                                   *
//*********************************************************************
//MVSCP    JOB
//*********************************************************************
//*                                                                   *
//*  THE MVSCP INPUT PARAMETER STRING MUST BE SPECIFIED ON THE        *
//*  EXEC STATEMENT. SET THE PARM TO 'IODEF' IN ORDER TO BUILD AN     *
//*  I/O CONFIGURATION.                                               *
//*                                                                   *
//*********************************************************************
//MVSCP    EXEC  PGM=CBPMVSCP,PARM='VALIDATE'
//*********************************************************************
//*                                                                   *
//*  THE SYS1.LINKLIB DATASET ON THE STEPLIB DD STATEMENT MUST        *
//*  BE THE SYS1.LINKLIB FOR THE SYSTEM THAT WILL USE THE I/O         *
//*  CONFIGURATION. THIS SYS1.LINKLIB DATASET CONTAINS THE CORRECT    *
//*  VERSION OF THE MVSCP MODULES.                                    *
//*                                                                   *
//*  THE APPROPRIATE UNIT AND VOLSER FOR THE SYS1.LINKLIB             *
//*  DATASET MUST BE SPECIFIED ON THE STEPLIB DD STATEMENT.           *
//*                                                                   *
//*********************************************************************
//STEPLIB  DD    DSN=SYS1.LINKLIB,DISP=SHR,
//             UNIT=XXXX,VOL=SER=XXXXXX
//SYSPRINT DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=80,BLKSIZE=1600)
```

Figure 8-3 (Part 1 of 2). JCL Statements in SYS1.SAMPLIB(MVSCPJCL)

```
//*********************************************************************
//*                                                                   *
//*  THE INPUT I/O DEFINITION DATASET MUST BE SPECIFIED ON THE        *
//*  SYSIN DD STATEMENT. THIS IS THE DATASET THAT CONTAINS YOUR       *
//*  IODEVICE AND UNITNAME STATEMENTS.                                *
//*                                                                   *
//*********************************************************************
//SYSIN    DD    DSN=XXXXXXXX,DISP=SHR
//*********************************************************************
//*                                                                   *
//*  SYSLMOD MUST POINT TO THE SYS1.NUCLEUS DATASET FOR THE SYSTEM    *
//*  THAT WILL USE THE I/O CONFIGURATION.                             *
//*                                                                   *
//*  SYSLIB MUST POINT TO THE SYS1.MACLIB DATASET FOR THE SYSTEM      *
//*  THAT WILL USE THE I/O CONFIGURATION.                             *
//*                                                                   *
//*  THE APPROPRIATE UNIT AND VOLSER MUST BE SPECIFIED ON THE         *
//*  SYSLMOD AND SYSLIB DD STATEMENTS.                                *
//*                                                                   *
//*********************************************************************
//SYSLMOD  DD    DSN=SYS1.NUCLEUS,DISP=SHR,
//             UNIT=XXXX,VOL=SER=XXXXXX
//SYSLIB   DD    DSN=SYS1.MACLIB,DISP=SHR,
//             UNIT=XXXX,VOL=SER=XXXXXX
//SYSUT1   DD    UNIT=SYSDA,SPACE=(TRK,(15,10))
//SYSUT2   DD    UNIT=SYSDA,SPACE=(TRK,(15,10)),DCB=(RECFM=FB,LRECL=80,
//             BLKSIZE=1680)
//SYSUT3   DD    UNIT=SYSDA,SPACE=(TRK,(30,10))
//SYSASMP  DD    SYSOUT=*
//SYSLOUT  DD    SYSOUT=*
//SYSLKEDP DD    SYSOUT=*
//SYSRPT   DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=2926)
//*********************************************************************
//*                                                                   *
//*  IF THE JES3 INITIALIZATION STREAM CHECKER DATA IS DESIRED,       *
//*  THEN THE OUTPUT DATASET FOR THE JES3 INITIALIZATION STREAM       *
//*  CHECKER DATA MUST BE SPECIFIED ON THE SYSJES3 DD STATEMENT.      *
//*  OTHERWISE, THE SYSJES3 DD STATEMENT MUST SPECIFY A DATASET       *
//*  NAME OF "NULLFILE".                                              *
//*                                                                   *
//*********************************************************************
//SYSJES3  DD    DSN=NULLFILE,DISP=SHR
```

Figure 8-3 (Part 2 of 2). JCL Statements in SYS1.SAMPLIB(MVSCPJCL)

Modifying SYS1.SAMPLIB(MVSCPXA) JCL

Comment statements throughout the SAMPLIB JCL identify changes that you must make in order to use the JCL statements. The following table summarizes those changes:

JCL Statement	Description and Changes Required
//MVSCP EXEC	The PARM= parameter, as coded in the sample, selects VALIDATE mode. To select another mode, change the PARM= parameter. For example, to select IODEF mode, code PARM='IODEF'.
//STEPLIB DD	Specify the UNIT= and VOL= parameter values. This DD statement must identify the SYS1.LINKLIB data set that contains the level of the MVS configuration program that corresponds to the level of the system on which the I/O configuration will be used. For example, if you are going to use the configuration on MVS/SP3.1.0, this DD must identify the SYS1.LINKLIB that contains the MVS/SP3.1.0 level of the MVS configuration program.
//SYSIN DD	Specify the DSN= parameter value. You must specify the name of the data set that contains the MVS configuration program input statements.
//SYSLMOD DD	Specify the UNIT= and VOL= parameter values. This DD must identify the data set where the MVS configuration program is to write the I/O configuration, that is, the SYS1.NUCLEUS data set of the system on which the configuration will be used.
//SYSLIB DD	Specify the UNIT= and VOL= parameter values. This DD statement must identify the SYS1.MACLIB of the system on which the configuration will be used. For example, if you intend to use this configuration on MVS/SP3.1.0, the DD statement must identify the MVS/SP3.1.0 level of SYS1.MACLIB.
//SYSJES3 DD	If you want the MVS configuration program to write data that will be read by the JES3 initialization stream checker, specify, on the DSN= parameter, the name of the data set where you want the data to be written. Otherwise, ignore this DD statement.

SYS1.SAMPLIB(MVSCP370) JCL Statements

To run the MVS configuration program on MVS/370, use the SYS1.SAMPLIB(MVS370) JCL. The SAMPLIB JCL contains the same DDs as the cataloged procedure (the use of the data sets allocated by those DDs is explained in Figure 8-2 on page 8-5) plus a STEPLIB DD that defines concatenated SYS1.LINKLIB data sets, a SYSUT4 DD, and a SYSUT5 DD.

The first STEPLIB DD must define the SYS1.LINKLIB that contains the MVS/370 system modules. The second STEPLIB DD must define the SYS1.LINKLIB that contains MVS configuration program modules that are at the same release level as the system on which the I/O configuration will be used.

Before using the SAMPLIB JCL, however, you must modify some of the JCL statements. For instructions on how to modify those statements, see "Modifying SYS1.SAMPLIB(MVSCP370) JCL Statements" on page 8-12.

```
//*********************************************************************
//*                                                                   *
//*  THIS JCL SHOULD BE USED TO RUN THE MVSCP ON AN MVS/370 SYSTEM.   *
//*                                                                   *
//*  IF YOU ARE RUNNING THE MVSCP ON AN EARLIER MVS/XA RELEASE        *
//*  LEVEL THAN THE LEVEL ON WHICH THE I/O CONFIGURATION WILL BE      *
//*  USED, THEN USE THE SAMPLE JCL IN SYS1.SAMPLIB(MVSCPXA).          *
//*                                                                   *
//*  IF YOU ARE RUNNING THE MVSCP ON AN MVS/XA RELEASE LEVEL THAT     *
//*  IS THE SAME AS THE RELEASE LEVEL ON WHICH THE I/O CONFIGURATION  *
//*  WILL BE USED, THEN USE THE CATALOGED PROCEDURE THAT IS IN        *
//*  SYS1.PROCLIB(MVSCP).                                             *
//*                                                                   *
//*********************************************************************
//MVSCP    JOB
//*********************************************************************
//*                                                                   *
//*  THE MVSCP INPUT PARAMETER STRING MUST BE SPECIFIED ON THE        *
//*  EXEC STATEMENT. SET THE PARM TO 'IODEF' IN ORDER TO BUILD AN     *
//*  I/O CONFIGURATION.                                               *
//*                                                                   *
//*********************************************************************
//MVSCP    EXEC  PGM=CBPMVSCP,PARM='VALIDATE'
//*********************************************************************
//*                                                                   *
//*  THE FIRST SYS1.LINKLIB DATASET ON THE STEPLIB DD STATEMENT       *
//*  IS THE SYS1.LINKLIB FOR THE SYSTEM ON WHICH THIS JCL IS BEING    *
//*  RUN. THIS SYS1.LINKLIB DATASET CONTAINS THE MVS/370 VERSION      *
//*  OF THE LOADER.                                                   *
//*                                                                   *
//*  THE SECOND SYS1.LINKLIB DATASET ON THE STEPLIB DD STATEMENT      *
//*  MUST BE THE SYS1.LINKLIB FOR THE SYSTEM THAT WILL USE THE I/O    *
//*  CONFIGURATION. THIS SYS1.LINKLIB DATASET CONTAINS THE CORRECT    *
//*  VERSION OF THE MVSCP MODULES.                                    *
//*                                                                   *
//*  THE APPROPRIATE UNIT AND VOLSER FOR THE SECOND SYS1.LINKLIB      *
//*  DATASET MUST BE SPECIFIED ON THE STEPLIB DD STATEMENT.           *
//*                                                                   *
//*********************************************************************
```

Figure 8-4 (Part 1 of 2). JCL Statements in SYS1.SAMPLIB(MVSCP370)

```
//STEPLIB  DD    DSN=SYS1.LINKLIB,DISP=SHR
//         DD    DSN=SYS1.LINKLIB,DISP=SHR,
//            UNIT=XXXX,VOL=SER=XXXXXX
//SYSPRINT DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=80,BLKSIZE=1600)
//*********************************************************************
//*                                                                   *
//*  THE INPUT I/O DEFINITION DATASET MUST BE SPECIFIED ON THE        *
//*  SYSIN DD STATEMENT. THIS IS THE DATASET THAT CONTAINS YOUR       *
//*  IODEVICE AND UNITNAME STATEMENTS.                                *
//*                                                                   *
//*********************************************************************
//SYSIN    DD    DSN=XXXXXXXX,DISP=SHR
//*********************************************************************
//*                                                                   *
//*  SYSLMOD MUST POINT TO THE SYS1.NUCLEUS DATASET FOR THE SYSTEM    *
//*  THAT WILL USE THE I/O CONFIGURATION.                             *
//*                                                                   *
//*  SYSLIB MUST POINT TO THE SYS1.MACLIB DATASET FOR THE SYSTEM      *
//*  THAT WILL USE THE I/O CONFIGURATION.                             *
//*                                                                   *
//*  THE APPROPRIATE UNIT AND VOLSER MUST BE SPECIFIED ON THE         *
//*  SYSLMOD AND SYSLIB DD STATEMENTS.                                *
//*                                                                   *
//*********************************************************************
//SYSLMOD  DD    DSN=SYS1.NUCLEUS,DISP=SHR,
//            UNIT=XXXX,VOL=SER=XXXXXX
//SYSLIB   DD    DSN=SYS1.MACLIB,DISP=SHR,
//            UNIT=XXXX,VOL=SER=XXXXXX
//SYSUT1   DD    UNIT=SYSDA,SPACE=(TRK,(15,10))
//SYSUT2   DD    UNIT=SYSDA,SPACE=(TRK,(15,10)),DCB=(RECFM=FB,LRECL=80,
//            BLKSIZE=1680)
//SYSUT3   DD    UNIT=SYSDA,SPACE=(TRK,(30,10))
//SYSUT4   DD    UNIT=SYSDA,SPACE=(TRK,(15,10))
//SYSUT5   DD    UNIT=SYSDA,SPACE=(TRK,(15,10))
//SYSASMP  DD    SYSOUT=*
//SYSLOUT  DD    SYSOUT=*
//SYSLKEDP DD    SYSOUT=*
//SYSRPT   DD    SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=2926)
//*********************************************************************
//*                                                                   *
//*  IF THE JES3 INITIALIZATION STREAM CHECKER DATA IS DESIRED,       *
//*  THEN THE OUTPUT DATASET FOR THE JES3 INITIALIZATION STREAM       *
//*  CHECKER DATA MUST BE SPECIFIED ON THE SYSJES3 DD STATEMENT.      *
//*  OTHERWISE, THE SYSJES3 DD STATEMENT MUST SPECIFY A DATASET       *
//*  NAME OF "NULLFILE".                                              *
//*                                                                   *
//*********************************************************************
//SYSJES3  DD    DSN=NULLFILE,DISP=SHR
```

Figure 8-4 (Part 2 of 2). JCL Statements in SYS1.SAMPLIB(MVSCP370)

Modifying SYS1.SAMPLIB(MVSCP370) JCL Statements

Comment statements throughout the SAMPLIB JCL identify changes that you must make in order to use the JCL statements. The following table summarizes those changes:

JCL Statement	Description and Changes Required
//MVSCP EXEC	The PARM= parameter, as coded in the sample, selects VALIDATE mode. To select another mode, change the PARM= parameter. For example, to select IODEF mode, code PARM='IODEF'.
//STEPLIB DD	Specify the UNIT= and VOL= parameter values on the second STEPLIB DD. This DD statement must identify the SYS1.LINKLIB data set that contains the level of the MVS configuration program that corresponds to the level of the system on which the I/O configuration will be used. For example, if you are going to use the configuration on MVS/SP3.1.0, this DD must identify the SYS1.LINKLIB that contains the MVS/SP3.1.0 level of the MVS configuration program.
//SYSIN DD	Specify the DSN= parameter value. You must specify the name of the data set that contains the MVS configuration program input statements.
//SYSLMOD DD	Specify the UNIT= and VOL= parameter values. This DD must identify the data set where the MVS configuration program is to write the I/O configuration, that is, the SYS1.NUCLEUS data set of the system on which the configuration will be used.
//SYSLIB DD	Specify the UNIT= and VOL= parameter values. This DD statement must identify the SYS1.MACLIB of the system on which the configuration will be used. For example, if you intend to use this configuration on MVS/SP3.1.0, the DD statement must identify the MVS/SP3.1.0 level of SYS1.MACLIB.
//SYSJES3 DD	If you want the MVS configuration program to write data that will be read by the JES3 initialization stream checker, specify, on the DSN= parameter, the name of the data set where you want the data to be written. Otherwise, ignore this DD statement.

Identifying the SYSIN and JES3 Data Sets

The SYSIN data set contains the input statements that define your I/O configuration. This required data set is defined by the //SYSIN DD statement in both the IBM-provided cataloged procedure and the IBM-provided SAMPLIB JCL.

The JES3 data set is required only if you want the MVS configuration program to write data that can be read by the JES3 initialization stream checker. The //SYSJES3 DD statement in both the IBM-provided cataloged procedure and the IBM-provided SAMPLIB JCL defines the JES3 data set.

SYSIN Data Set

Each time you invoke the MVS configuration program, you *must* specify the name of the SYSIN data set. There is no default for this name.

If you use the IBM-provided cataloged procedure, specify the SYSIN data set name on the DEFDATA parameter of the PROC statement.

If you use the IBM-provided SAMPLIB JCL, specify the SYSIN data set name on the DSN= parameter of the SYSIN DD statement.

JES3 Data Set

The JES3 data set is optional. Specify its name only if you want the MVS configuration program to write data that the JES3 initialization stream checker can read. The default name is NULLFILE which causes no data to be written.

Note: The JES3 data will be written only if:

- You specify the name of the JES3 data set.
- You select either IODEF mode or EDT mode.
- The MVS configuration program detects no errors during execution.

If you use the IBM-provided cataloged procedure and wish to specify the JES3 data set name, use the JES3 parameter on the PROC statement.

If you use the IBM-provided SAMPLIB JCL and wish to specify the JES3 data set name, use the DSN parameter on the SYSJES3 DD statement.

The MVS configuration program can write the JES3 data to either a sequential data set or to a member of a partitioned data set. JES3, however, requires that the data be in a partitioned member. JES3 also requires that the member name be the same as the processor name defined in the JES3 initialization stream. For additional information about how JES3 uses this data set, see *JES3 Initialization and Tuning*.

Ordering Statements in the Input Stream

To provide an order to the input stream, IBM recommends that you group the statements as shown in the following example.

```
IOCONFIG
ID
CHPID
CNTLUNIT
IODEVICE
     .
     .
IODEVICE
EDT
UNITNAME
     .
     .
UNITNAME
NIPCON
```

Note: Do not include an **END** statement in the input stream. An **END** statement causes the MVS configuration program to abnormally terminate.

This grouping offers these advantages:

- The I/O configuration identifier appears on the first statement. This might make it easier to locate a specific configuration.
- The IOCP statements are grouped together as are the IODEVICE statements. This grouping might make it easier to locate specific statement types when you edit the input stream.

Required Order for EDT and UNITNAME Statements

The only statements that must appear in a specific order are the **EDT** statements, and **UNITNAME** statements.

One Eligible Device Table

If you define only one eligible device table, you can either include or omit the **EDT** statement. If you include an **EDT** statement, place it ahead of all **UNITNAME** statements. If you omit the **EDT** statement, IBM recommends you place all **UNITNAME** statements after the last **IODEVICE** statement.

Multiple Eligible Device Tables

If you define multiple eligible device tables in one input stream, include one **EDT** statement for each eligible device table. Place each **EDT** statement immediately ahead of the **UNITNAME** statements that correspond to that eligible device table.

Selecting the Mode of Operation

The MVS configuration program has four modes of operation: IODEF, EDT, VALIDATE (the default), and DEBUG.

The mode you select will depend on what you want to do. Use the following decision table to determine the mode to select.

Figure 8-5 (Page 1 of 2). Selecting the Mode of Operation

To do this task:	Select	The MVS Configuration Program Will:
Migrate an I/O configuration from an earlier MVS system.	IODEF	• Validity check the input statements and issue error messages for detectable errors. • If execution is successful, writes new I/O configuration members, issues reports, and optionally writes configuration data for use by the JES3 initialization stream checker.
Update an existing I/O configuration by adding, deleting, or changing one or more of the following statements: • IOCONFIG • IODEVICE • EDT • UNITNAME • NIPCON **Note:** If you add, delete, or change EDT or UNITNAME statements only, you can select EDT mode instead of IODEF mode. In EDT mode, the MVS configuration program replaces only the affected EDT members of the configuration. In IODEF mode, the MVS configuration program replaces all configuration members.	IODEF	• Validity check the input statements and issue error messages for detectable errors. • If execution is successful, replaces the existing I/O configuration members, issues reports, and optionally writes configuration data for use by the JES3 initialization stream checker.
Update an existing configuration by adding, deleting, or changing one or more of the following statements: • EDT • UNITNAME **Note:** If you change any other statements, you must select IODEF mode.	EDT	• Validity check the input statements and issue error messages for detectable errors. • If execution is successful, replaces the affected EDT members of the configuration, issues reports, and optionally writes configuration data for use by the JES3 initialization stream checker.
Define a new I/O configuration,	IODEF	• Validity check the input statements and issue error messages for detectable errors. • If execution is successful, writes new I/O configuration members, issues reports, and optionally writes configuration data for use by the JES3 initialization stream checker.

Figure 8-5 (Page 2 of 2). Selecting the Mode of Operation

To do this task:	Select	The MVS Configuration Program Will:
Validity check input statements without creating or updating a configuration,	VALIDATE	• Validity check the input stream and issue error messages for detectable errors. • Issue the same reports that are issued in IODEF mode. **Note:** The MVS configuration program does not build an I/O configuration.
Gather information to debug any of the following problems: • The assembler parsing macros are generating incorrect code • The loader is having problems loading records • EDT problems • The MVS configuration program ESTAE is masking a problem (ESTAE coverage is disabled in DEBUG mode, thus it cannot mask the problem)	DEBUG	• Validity check the input statements and issue messages for detectable errors. • Write the assembler text records, loader output, and certain MVS configuration program control blocks data sets. **Note:** The MVS configuration program *does not* build the I/O configuration.

How you select the mode depends on whether you use the IBM-provided catalog procedure or the IBM-provided SAMPLIB JCL.

- If you use the cataloged procedure, select the mode by using the TYPE= parameter on the PROC statement.
- If you use the SAMPLIB JCL, select the mode by using the PARM= parameter on the EXEC statement.

Invoking the MVS Configuration Program

There are two ways to invoke the MVS configuration program, dynamically or with JCL statements.

JCL Invocation

If you are using the IBM-provided SAMPLIB JCL, modify the JCL and then submit it as you would any other job.

To invoke the MVS configuration program through the IBM-provided cataloged procedure, use JCL similar to that shown in the following example. For information on how to use the invocation JCL to override statements in the cataloged procedure, see *JCL Reference* and *JCL Users Guide*.

```
//MVSCP    JOB 'ACCTINFO','NAME',MSGLEVEL=(1,1),
//              MSGCLASS=R,REGION=4096K
//TEST     EXEC  PROC=MVSCP,TYPE='IODEF',
//              DEFDATA='INSTALL.IODEF',
//              JES3='INSTALL.JES3.DATA'
```

Dynamic Invocation

To dynamically invoke the MVS configuration program, your program can either issue the ATTACH macro or the LINK macro. The entry point name of the MVS configuration program is CBPMVSCP.

Register Content

Upon entry into CBPMVSCP, the registers must contain the following:

Register	Content
R0	Any value
R1	The address of a two-word parameter list
R2 - R12	Any value
R13	The address of an 18-word register save area
R14	The return address
R15	The entry point address of CBPMVSCP

Parameter List

The two-word parameter list pointed to by register 1 must start on a fullword boundary. The list must contain:

word 1	A pointer to a variable length required options list. If you do not use an optional list of alternate DD names (that is, word 2 is binary zeros) set the high-order bit to 1, identifying the end of the parameter list.
word 2	A pointer to an optional list of alternate DD names.

Options List

This list, pointed to by word 1, must start on a halfword boundary that is not also a fullword boundary.

- The first two bytes must contain a binary count of the number of bytes in the rest of the list.
- The rest of the list must identify the MVS configuration program option that is to be used. You can identify only one option per invocation. Valid values, which must be in character string format, are:

  IODEF
  EDT
  VALIDATE
  DEBUG

Alternate DD Names List

If used, this optional list, pointed to by word 2, must start on a halfword boundary that is not also a fullword boundary.

- The first two bytes must contain a binary count of the number of bytes in the rest of the list.
- The rest of the list specifies alternate DD names that you wish to use in place of the standard DD names. For a list of standard DD names, see "Standard DD Name List" on page 8-19.
  - DD names in the alternate list must appear in the same sequence as they appear in the standard list.
  - Each name must be eight characters long. If a name contains fewer than eight characters, pad it with blanks.
  - If you omit an alternate DD name, set that entry in the alternate DD names list to binary 0.
  - Entry's in the alternate DD name list that correspond to empty entry's in the standard DD name list must be set to binary 0.

Standard DD Name List

The standard DD names list contains the following DD names in the order shown. Each entry, or slot, is eight bytes long. Names are left justified, and if shorter than eight bytes, padded with blanks. Empty slots contain binary 0.

DD Name	Notes
0	Empty slot.
0	Empty slot.
SYSLMOD	Contains the I/O configuration built by the MVS configuration program. (This is the linkage editor SYSLMOD file.)
SYSLIB	Contains the MVS configuration program parsing macros. (This is the assembler SYSLIB file.)
SYSIN	Contains the MVS configuration program input statements.
SYSPRINT	Contains messages issued by the MVS configuration program.
0	Empty slot.
SYSUT1	Used by the assembler and the linkage editor as a work file.
SYSUT2	The MVS configuration program, before processing its input stream, writes the input stream to this file. The file then becomes the assembler SYSIN file.
SYSUT3	This file serves several purposes: • It is the assembler SYSPUNCH file: the assembler writes its output to this file which then becomes the loader SYSLIN file. • It is the linkage editor SYSLIN file: the MVS configuration program writes the I/O configuration object modules to this file which then becomes the linkage editor SYSLIN file.
0	Empty slot.
0	Empty slot.
0	Empty slot.
0	Empty slot.
0	Empty slot.
SYSASMP	Contains the listing written by the assembler. (This is the assembler SYSPRINT file.)
SYSLOUT	Contains messages issued by the loader. (This is the loader SYSLOUT file.) The loader writes this file only when the MVS configuration program runs in DEBUG mode.
SYSLKEDP	Contains linkage editor diagnostic output. (This is the linkage editor SYSPRINT file.) The linkage editor writes this file only when the MVS configuration program runs in IODEF mode or EDT mode.
SYSRPT	Contains the MVS configuration program reports: • The IOS report. • The allocation report. • The NIP report.
SYSJES3	Contains data, written by the MVS configuration program, that will be read by the JES3 initialization stream checker.
SYSUT4	The assembler SYSUT2 workfile. Used only if you execute the MVS configuration program on MVS/370.
SYSUT5	The assembler SYSUT5 workfile. Used only if you execute the MVS configuration program on MVS/370.

Verifying MVS Configuration Program Output

After the MVS configuration program executes, check for successful execution by verifying the output. To verify the output, check the JES job log for system messages, check the SYSPRINT data set for MVS configuration program messages, ensure that you received all expected output, and check the reports in the SYSRPT data set against the planned I/O configuration.

1. **Check the JES job log:**

   - If the messages indicate a JCL error:
     - If the error is in installation-provided JCL, correct the error and rerun the job.
     - If the error is in IBM-provided JCL (the MVS configuration program cataloged procedure or one of the two SAMPLIB members), refer to the *Component Diagnosis: MVSCP*.
   - **Look for message IEF142I:**
     Message IEF142I, If present, indicates that the MVS configuration program ran to completion. The IEF142I condition code corresponds to the MVS configuration program return code. Check the MVS configuration program messages and return code in the SYSPRINT data set. For an explanation of messages and return codes, see "MVS Configuration Program Messages and Return Codes" on page 10-2.
     - If the MVS configuration program messages indicate a successful run, go to step 2 on page 8-21.
     - If the MVS configuration program messages identify an error that you should be able to correct, such as an input stream error, make the correction and rerun the job. If after the correction the error persists, refer to the *Component Diagnosis: MVSCP*.
     - If the MVS configuration program messages indicate an error that you are unable to correct, for example, an MVS configuration program logic error, refer to the *Component Diagnosis: MVSCP*.
   - **If the JES job log or a dump indicates system abend 088:**
     The reason code that accompanies the abend code identifies the reason for the error. There will also be a corresponding MVS configuration program message that further explains the error.
     - MVS configuration program messages are documented in "MVS Configuration Program Messages and Return Codes" on page 10-2.
     - If you ran the MVS configuration program on MVS/370, for abend codes and reason codes see *OS/VS Message Library: VS2 System Codes*.
     - If you ran the MVS configuration program on MVS/ESA, see *System Codes*.
     - If you ran the MVS configuration program on MVS/XA, for abend codes and reason codes see *MVS/Extended Architecture Message Library: System Codes*.
     - If the messages provide enough information for you to correct the error, for example, message CBP148I indicates an incorrect LRECL specification in one of your DD statements, make the correction and rerun the job. If, after the correction, the error persists, refer to the *Component Diagnosis: MVSCP*.

- If the messages provide insufficient information for you to correct the error, refer to the *Component Diagnosis: MVSCP*.

- **If the JES job log or a dump indicates a system abend other than 088:** Refer to the *Component Diagnosis: MVSCP*.

2. **Check for correct output:**
   Figure 8-6 identifies, for each mode (IODEF, EDT, VALIDATE, and DEBUG), the output that the MVS configuration program produces when it executes successfully and the ddname that defines the data set to which the output is written. An X identifies expected output.

   If output is missing, refer to the *Component Diagnosis: MVSCP*.

Figure 8-6. Normal MVS Configuration Program Output

Output	IODEF	EDT	VALIDATE	DEBUG
IOS report (SYSRPT)	X		X	X
Allocation report (summary) (SYSRPT)	X	X	X	
Allocation report (full report) (SYSRPT)				X (See note 2a on page 8-22.)
NIP report (SYSRPT)	X		X	X
MVSCP messages (SYSPRINT)	X (See note 2b on page 8-22.)	X (See note 2b on page 8-22.)	X (See note 2c on page 8-22.)	X (See note 2c on page 8-22.)
Return code (SYSPRINT)	X	X	X	X
Input statement listing (summary) (SYSASMP)	X	X	X	
Input statement list (full listing) (SYSASMP)				X
Parsing macro listing (SYSASMP)				X
Assembler RLD (SYSASMP)				X
Assembler cross-reference listing (SYSASMP)				X
Assembler ESD (SYSASMP)				X
Loader map (SYSLOUT)				X
Linkage editor map (SYSLKEDP)	X (See note 2e on page 8-22.)	X (See note 2f on page 8-22.)		
JES3 data (SYSJES3)	X (See note 2d on page 8-22.)	X (See note 2d on page 8-22.)		
IOSUCBxx member (SYSLMOD)	X			
IOSIITxx member (SYSLMOD)	X			
IEFEDTxx member (SYSLMOD)	X	X		
IEANCTxx member (SYSLMOD)	X			

Notes:

a. The full allocation report consists of:

- Lookup value section.
- Generic section.
- Group section.
- Group pointer section.
- Group descriptor section.
- Device number section.
- Group mask section.
- Device number summary.

b. The MVS configuration program issues CBP149I. If JES3 output was requested and successfully written, the MVS configuration program also issues CBP600I. The MVS configuration program may also issue other messages depending on the conditions it detects.

c. The MVS configuration program issues CBP144I. The MVS configuration program may also issue other messages depending on the conditions it detects.

d. The MVS configuration program writes JES3 data only if the SYSJES3 data set has been allocated and the name of the JES3 data set has been specified in the invocation JCL.

e. The linkage editor map should indicate that members IOSUCBxx, IOSIITxx, IEFEDTxx, and IEANCTxx have been replaced.

f. The linkage editor map should indicate that member IEFEDTxx has been replaced.

3. **Review the reports:**
   If you have received all expected output, review the MVS configuration program reports (For an explanation of how to read the reports, see Chapter 9, "Reading MVS Configuration Program Reports" on page 9-1). If information is missing or is incorrect, refer to the *Component Diagnosis: MVSCP*.

4. **Refer to the** *Component Diagnosis: MVSCP* **:.**
   If the MVS configuration program messages and output appear to be correct but configuration-related errors occur when the installation uses the configuration, refer to the *Component Diagnosis: MVSCP*.

Chapter 9. Reading MVS Configuration Program Reports

The MVS configuration program writes three reports:

- An I/O supervisor (IOS) report.
- An allocation report.
- A nucleus initialization program (NIP) report.

The MVS configuration program writes the reports when it runs in either IODEF mode, EDT mode, or VALIDATE mode, and it issues a return code no greater than 4.

The MVS configuration programs writes the reports to the //SYSRPT data set. The reports contain the following information:

Report	Content
IOS	Identifies by device number, unit, and model each device included in the configuration. Each line in the report represents a unique device number within that configuration. The report also includes for each device: • The base device number (base exposure) if the device is a multiple exposure device. • The names of the error recovery program, device descriptor table, module list table (identifies device support modules), and unit information module (UIM) for the device. • The status of specific UCB fields for the device.
Allocation	Contains the following information for each generic device type and for each esoteric device group: • The group name and whether it is a generic group or an esoteric group. • Whether the group is eligible for VIO. • Any generics or esoterics that the MVS configuration program automatically generates. • The affinity index assigned to the device type. • Data that corresponds to the UCB field UCBTYP. • A list of the device numbers that the group includes. The report organizes the numbers to show how they are mapped into different allocation groups.
NIP	Identifies by device number, unit, and model the devices that NIP will be allowed to use as consoles. The report lists the devices in the order NIP will try to use them.

Locating Information in this Chapter

This chapter contains the following major topics:

> **Note:** The acronym MVSCP is used in the reports and other computer-generated output to refer to the MVS configuration program.

Requesting Reports and Validating the Input Stream

You can request the reports and validate the input stream without building an I/O configuration. You may want to do this, for example, if you have lost the reports for an existing configuration.

To request the reports, run the MVS configuration program in VALIDATE mode. The SYSIN data set must contain the input statements that define the configuration for which you want the report.

An example of an IOS report appears on page 9-4.

This page intentionally left blank.

Reading the IOS Report

```
MVSCP LEVEL  1.0   IOCONFIG ID=05            IOS CONTROL BLOCK REPORT              TIME: 14.12  DATE: 04/07/86  PAGE      1

      (1) IOCP  MSG1 = 'I/O CONFIGURATION 05                                        '
          IOCP  MSG2 = '                                                            '

DEV#  STMT    UNIT    MODL   UCBTYP   BASE ERP-NAME  DDT-NAME  MLT-NAME  OPT  UIM-NAME  ATI  AL  SH  SW  MX  MI
----  ------  --------  ----  --------  ---- --------  --------  --------  ---  --------  ---  --  --  --  --  --

 00C     15 2540R    1     10000801       IGE0001C  IECVDDT5  IEAMLT34  (11)  CBPUS012    0              Y
 00E     16 3800           1000080E       IGE0000G  IGGDDTP1  IEAMLT09        CBPUS011    0
 011     17 3800           1000080E       IGE0000G  IGGDDTP1  IEAMLT09        CBPUS011    0
 140      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 141      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y

 142      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 143      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 144      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 145      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 146      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y

 147      3 2305     2     30502007  140  IECVDERP  IGGDDT05  IEAMLT10        CBPUS013    0                  Y
 190      4 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 191      4 3380           3010200E  (7)  IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 220      5 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 221      5 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0

 222      5 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 223      5 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 3E0     13 3277     2     12001009       IGE0010E  IECVDDT4  IEAMLT05        CBPUS004   28
 410      6 3350           3050200B       IECVDERP  IGGDDT04  IEAMLT03        CBPUS001    0
 411      6 3350           3050200B       IECVDERP  IGGDDT04  IEAMLT03        CBPUS001    0

 440      7 3330     1     30502009       IECVDERP  IGGDDT30  IEAMLT11        CBPUS001    0              Y
 441      7 3330     1     30502009       IECVDERP  IGGDDT30  IEAMLT11        CBPUS001    0              Y
 450     11 3480           78008080       IGE0001E  DDTR3480  IEAMLT08        CBPUS005    0              Y
 4E2     14 3270     X     12001009       IGE0010E  IECVDDT4  IEAMLT05        CBPUS004   28
 570     12 3420     3     34008003       IGE0000I  DDTR3400  IEAMLT06        CBPUS005    0              Y

 640      8 3330    (5)    30502009       IECVDERP  IGGDDT30  IEAMLT11        CBPUS001    0              Y
 A40      9 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 A41      9 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 A42      9 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 A43      9 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0

 F20     10 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 F21     10 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 F22     10 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
 F23     10 3380           3010200E       IECVDERP  IGGDDT01  IEAMLT02        CBPUS002    0
  (2)    (3) (4)            (6)            (8)       (9)       (10)            (12)        (13)

MVSCP LEVEL  1.0   IOCONFIG ID=05            IOS CONTROL BLOCK REPORT              TIME: 14.12  DATE: 04/07/86  PAGE      2

   (14) KEY          KEY DESCRIPTION
        ---          ---------------

        DEV#      -  DEVICE NUMBER
        STMT      -  STATEMENT WHICH DEFINED THIS DEVICE
        UNIT      -  DEVICE NAME
        UCBTYP    -  UCB TYPE BYTES
        BASE      -  BASE DEVICE NUMBER FOR MULTIPLE EXPOSURE DEVICES
        ERP-NAME  -  ERROR RECOVERY PROGRAM
        DDT-NAME  -  DEVICE DESCRIPTOR TABLE
        MLT-NAME  -  MODULE LIST TABLE
        OPT       -  OPTIONAL MLT INDICATOR
        UIM-NAME  -  UNIT INFORMATION MODULE

        ATI       -  ATTENTION TABLE INDEX (UCBATI)
        AL        -  ALTERNATE CONTROL UNIT (UCBALTCU)
        SH        -  SHARED UP OPTION (UCBSHRUP)
        SW        -  DEVICE CAN BE SWAPPED BY DDR (UCBSWAPF)
        MX        -  DEVICE HAS MULTIPLE EXPOSURES (UCBMTPXP)
        MI        -  MIH PROCESSING SHOULD BE BYPASSED (UCBMIHPB)

        0         -  MLT IS OPTIONAL
        Y         -  DEVICE SUPPORTS THIS FEATURE
        BLANK     -  DEVICE DOES NOT SUPPORT THIS FEATURE



              TOTAL NUMBER OF DEVICES BY CLASS
        (15)  ----- ------ -- ------- -- -----   (16)
        CLASS NAME              CLASS TYPE    DEVICE COUNT
        ----------              ----------    ------------

        TAPE                        80              2
        COMMUNICATION DEVICES       40              0
        C-T-C                       41              0
        DASD                        20             27
        GRAPHICS                    10              2
        UNIT RECORD                 08              3
        CHARACTER READERS           04              0

        TOTAL NUMBER OF I/O DEVICES DEFINED BY THIS I/O CONFIGURATION          34
```

Figure 9-1. IOS Report Example

1. Information from the MSG1 and MSG2 fields of the IOCP ID statement.
2. The device number assigned to the device.
3. The sequence number of the IODEVICE statement that defined the I/O device. The MVS configuration program numbers the statements sequentially starting with the first statement in the input stream. The Assembler prints each input statement along with its corresponding MVS configuration program sequence number.
4. The device type. This information comes from the IODEVICE statement UNIT parameter.
5. The model of the device. This information comes from the IODEVICE statement MODEL parameter.
6. The UCB type field. This information comes from the UCBTYP field of the device's UCB.
7. The base exposure of a multiple exposure device.
8. The name of the error recovery program that handles device errors for the defined device.
9. The name of the device descriptor table. This table identifies the modules that perform device dependent processing for the defined device.
10. The name of the module list table. This table identifies the device support modules for the defined device.
11. The module list table indicator. This field identifies optional module list tables. O indicates an optional table.
12. The name of the unit information module (UIM) that processed the IODEVICE statement for the specified device.
13. UCB information for the specified device. The UCB is described in the *Data Areas* book.

    ATI - (UCBATI): The attention table index.

    AL - (UCBALTCU): Y indicates that the device has an alternate control unit, a blank indicates that it does not.

    SH - (UCBSHRUP): Y indicates that the device is sharable in uniprocessor mode, a blank indicates that it is not.

    SW - (UCBSWAPF): Y indicates that the device can be swapped, a blank indicates that it cannot.

    MX - (UCBMTPXP): Y indicates that the device is a multiple exposure device, a blank indicates that it is not.

    MI - (UCBMIHPB): Y indicates that missing interrupt handler checking is to be bypassed, a blank indicates that it is not to be bypassed.
14. A key that briefly describes each area of data found in the report.
15. A summary of the devices defined in the configuration by device class. The device classes correspond to the classes defined in the UCB field UCBTBYT3.
16. The number of devices defined in each device class. Each device number is counted as a device.

Reading the Allocation Report

```
MVSCP LEVEL  1.0                    IEFEDT05 FORMAT - DEVICE NUMBER SUMMARY          TIME: 14.12  DATE: 04/07/86  PAGE      1

 UNIT NAME     NAME TYPE     VIO    AFFINITY    ALLOCATION      ASSOCIATED                      DEVICE NUMBER LIST
                                     INDEX      DEVICE TYPE     GENERICS
 ---------     ---------     ---    --------    -----------     ----------    ------------------------------------------------
 2305-2        GENERIC        (3)   FFFF        30502007                      140
 3380          GENERIC              FFFF        3010200E                      190 191 220 221 222 223 A40 A41 A42 A43 F20 F21
                                                                                  F22 F23
 3330-1        GENERIC              0002        3050200D                      410 411
 3330          GENERIC              0001        30502009                      440 441 640
 3480          GENERIC              0008        78008080                      450
 3400-3        GENERIC              0030        34008003                      570
 3800          GENERIC              FFFF        1000080E                      00E 011
 2540          GENERIC              FFFF        10000801                      00C
 3277-2        GENERIC              FFFF        12001009                      3E0 4E2
 2400-3        G/GENERIC            0030        34008003      3400-3          570
 3400-9        G/GENERIC            0008        78008080      3480            450
 SYSALLDA      G/ESOTERIC           FFFF        30502007      2305-2          140
                                                3010200E      3380            190 191 220 221 222 223 A40 A41 A42 A43 F20 F21
                                                                                  F22 F23
                                                3050200D      3330-1          410 411
                                                30502009      3330            440 441 640
  (1)           (2)                  (4)         (5)           (6)             (7)
```

Figure 9-2. Allocation Report Example

1. The name of the device group as it appeared on a UNITNAME statement NAME parameter.
2. Indicates whether this is a generic device group or an esoteric device group. G/GENERIC identifies a generic group that includes generated generics. G/ESOTERIC identifies a group that contains generated esoterics. The column labeled "ASSOCIATED GENERICS" identifies the included generics. The MVS configuration program includes generated generics so installations can still use older JCL input streams on the newer releases of MVS.
3. Indicates whether the group is eligible for VIO. Y indicates an eligible group. A blank indicates an ineligible group.
4. The affinity index for the device type. Used by MVS allocation programs to determine which devices have affinity to each other. Devices have affinity if either of the following statements are true:
   - The devices have the same affinity index
   - The affinity index for one of the devices is a subset of the other device's affinity index

   An affinity index is a subset of another if both of the following statements are true:
   - Neither index has a value of X'FFFF'
   - One or more bits in one index are set to binary one and one or more corresponding bits in the other index are also set to binary one.
5. The format and the meaning of the information in this field is the same as UCBTYP field in the UCB. For a description of the UCBTYP field, see the *Data Areas* book.
6. Identifies the generics that the group includes. If the column labeled "NAME TYPE" specifies G/GENERIC, this column, "ASSOCIATED GENERICS," identifies the generated generics.
7. The device numbers of the devices that are included in the group. Where a generic or esoteric group consists of more than one allocation group, and that group has too many device numbers to fit on one line, the indentation of device numbers at the start of the next line indicates a continuation of the group.

Reading the NIP Report

```
MVSCP LEVEL  1.0   IOCONFIG ID=05        NIP CONTROL BLOCK REPORT          TIME: 14.12  DATE: 04/07/86

                                  NIP CONSOLE DEVICES
                                  -------------------

                    DEVICE #          UNIT          MODEL
                    --------          --------      -----

                      4E2             3270            X
                      3E0             3277            2
                      (1)             (2)           (3)
```

Figure 9-3. NIP Report Example

1. The device numbers of the NIP console.
2. The device type. This information corresponds to the UNIT parameter on the IODEVICE statement.
3. The model. This information corresponds to the MODEL parameter on the IODEVICE statement.

Chapter 10. Debugging, Messages, and Codes

This book divides the problems you may encounter while using the MVS configuration program into two categories: input stream problems and MVS configuration program problems. Depending on the type of problem, you can receive one or more messages, an ABEND code and a reason code, or both. This chapter provides suggestions for debugging problems and contains the MVS configuration program messages and return codes.

Note: MVS configuration program abend codes and reason codes are documented in the following books:

- If you run the MVS configuration program on MVS/ESA, see *System Codes*.
- If you run the MVS configuration program on MVS/XA, see *MVS/XA Message Library: System Codes*.
- If you run the MVS configuration program on MVS/370, see *OS/VS Message Library: VS2 System Codes*.

Debugging Input Stream Problems

Input stream problems are usually caused by coding errors on the input statements, information omitted from the statements, omitted statements, or errors in the parameters you provide. You will receive messages that indicate the cause of the problem. You can receive messages from the assembler, the linkage editor, the loader, or the MVS configuration program. These messages appear in the data sets that are defined by the following DD statements:

Message Source	DD Name
assembler	SYSASMP
linkage editor	SYSLKEDP
loader	SYSLOUT
MVS configuration program	SYSPRINT

The SYSASMP data set also contains a listing of the input statements. A number that the MVS configuration program prints next to each statement and next to each message enables you to correlate messages to the affected input statement. To correct input stream problems, follow the instructions found in the message documentation.

If the SYSPRINT data set cannot be opened, a message will appear in the job listing.

Debugging the MVS Configuration Program

MVS configuration program problems can be caused by logic errors in either the MVS configuration program, (including the IBM-provided UIMs), logic errors in programs invoked by the MVS configuration program (assembler, linkage editor, and loader), or by logic errors in installation-provided UIMs. MVS configuration program problems are indicated by messages and, possibly, by an ABEND code and reason code.

If the MVS configuration program job ABENDS and the invocation JCL contains either a SYSABEND, SYSMDUMP, or SYSUDUMP DD statement, you will also receive a dump.

Depending on the type of problem the MVS configuration program encounters, DEBUG mode might provide additional information that can help you solve the problem. If you suspect any of the following problems, select DEBUG mode:

- The MVS configuration program parsing macros are generating incorrect code
- The loader is having a problem loading MVS configuration program records
- There are problems with EDT information
- The MVS configuration program ESTAE is masking a problem

To analyze MVS configuration program problems, use the message documentation and any dump information that you receive.

MVS Configuration Program Messages and Return Codes

Each time the MVS configuration program executes, it issues messages and a return code. Those messages and return codes are explained in this chapter.

In addition to the messages documented in this chapter, you may also receive messages from the Assembler, the Loader, or the Linkage Editor.

Note: Messages in the range CBP900 to CBP999 are reserved for installations to use when they write UIMs.

Message Format

Each MVS configuration program message has the following format:

STATEMENT	SEV	MSG ID	MESSAGE TEXT

STATEMENT

The assigned number of the input statement, if any, that caused the MVS configuration program to issue the message. The input statements and their numbers appear in the SYSASMP data set following the messages.

There may be multiple messages that pertain to one input statement.

SEV
The severity of the message

Severity	Meaning
I	Informational message. An informational message has no affect on MVS configuration program processing.
W	Warning message. A warning message has no affect on MVS configuration program processing. You should, however, read all of the warning messages issued to ensure that an acceptable I/O configuration was built.
E	Error message. The MVS configuration program performs a syntax check and semantic check of the entire input stream but does not build an I/O configuration.
T	Terminating message. The MVS configuration program abnormally terminates without processing the remaining input statements.

MESSAGE ID
A seven character alphanumeric value that uniquely identifies each message. The messages in this chapter are arranged in ascending order according to message identifiers.

MESSAGE TEXT
Information that summarizes the condition that caused the MVS configuration program to issue the message.

How This Book Presents Message Information

This book presents the messages in ascending numerical order. First it presents the message number and the message text. This is followed by:

Explanation
An explanation of the condition that caused the message to be issued.

System Action
A description of the way that the MVS configuration program handles the condition.

System Programmer Response
An explanation of what you should do to correct the condition or to further isolate the problem.

Note: In MVS configuration program messages, the acronym WTO refers to the write-to-operator macro.

What the Return Codes Mean

The MVS configuration program issues return codes 0, 4, 8, and 16. The return code issued corresponds to the highest severity level message issued.

Return Code	Highest Severity Level Message Issued:
0	Informational message
4	Warning message
8	Error message
16	Terminating message

Example of Message Output

The following figures show an example of a partial assembler listing of an MVS configuration program input stream and the messages that resulted from processing that stream. If a message refers to a specific input statement, the statement number that the MVS configuration program assigned appears to the left of the message. For example, in Figure 10-1, message CBP401I refers to statement 9. In Figure 10-2 on page 10-5, statement 9 corresponds to assembler statement number 2062.

MVS Configuration Program Messages

```
MVSCP LEVEL  1.0     IOCONFIG ID=05     TIME: 16.20  DATE: 02/21/86  PAGE      1

STATEMENT SEV MSG ID   MESSAGE TEXT

        9  E  CBP401I  12 IS NOT A VALID MODEL NUMBER FOR THE 3330 DEVICE

       11  E  CBP352I  FEATURE AUTOPOLL NOT RECOGNIZED FOR 3380 DEVICE

           W  CBP400I  THE 3851 AND 3330V DEVICES SHOULD BOTH BE SPECIFIED OR
                       OMITTED FROM THE I/O CONFIGURATION DEFINITION

           I  CBP145I  NO OUTPUT WRITTEN TO SYSLMOD DATA SET. VALIDATE
                       PROCESSING FORCED DUE TO ERROR

        3  I  CBP429I  REPLICATION FACTOR OF ONE IS USED. EIGHT DEVICE NUMBERS
                       DEFINED

TOTAL MESSAGES      INFORMATIONAL      WARNING      ERROR      TERMINATING
         5                  2             1          2              0

RETURN CODE WAS   8
```

Figure 10-1. MVS Configuration Program Message Example

Assembler Listing of Input Statements

```
       MVSCP INPUT LISTING                                                                    PAGE    2

LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                ASM H V 02 16.20 02/21/86
                                 1782 ********************************************************************** 00000300
                                 1783 *                      DASDS                                           * 00000400
                                 1784 ********************************************************************** 00000500
                                 1785 *
                                 1786          IODEVICE ADDRESS=(140,8),UNIT=2305,MODEL=2,CUNUMBR=021          00000600
                                 1787+*, 3 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 1788+                                                                         02-01442
                                 1830+                                                                         01-01254
                                 1831+                                                                         01-01255
                                 1832          IODEVICE ADDRESS=(190,2),UNIT=3380,CUNUMBR=023                  00001110
                                 1833+*, 4 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 1834+                                                                         02-01442
                                 1876+                                                                         01-01254
                                 1877+                                                                         01-01255
                                 1878          IODEVICE ADDRESS=(1A0,6),UNIT=3330V,CUNUMBR=024                 00000700
                                 1879+*, 5 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 1880+                                                                         02-01442
                                 1922+                                                                         01-01254
                                 1923+                                                                         01-01255
                                 1924          IODEVICE ADDRESS=(220,4),UNIT=3380,CUNUMBR=027                  00001110
                                 1925+*, 6 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 1926+                                                                         02-01442
                                 1968+                                                                         01-01254
                                 1969+                                                                         01-01255
                                 1970          IODEVICE ADDRESS=(410,2),UNIT=3330,MODEL=1,CUNUMBR=031          00000800
                                 1971+*, 7 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 1972+                                                                         02-01442
                                 2014+                                                                         01-01254
                                 2015+                                                                         01-01255
                                 2016          IODEVICE ADDRESS=(440,2),UNIT=3350,CUNUMBR=034                  00000900
                                 2017+*, 8 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 2018+                                                                         02-01442
                                 2060+                                                                         01-01254
                                 2061+                                                                         01-01255
                                 2062          IODEVICE ADDRESS=(610,2),UNIT=3330,MODEL=12,CUNUMBR=019         00001000
                                 2063+*, 9 IS THE PREVIOUS MVSCP STATEMENT NUMBER                              02-01441
                                 2064+                                                                         02-01442
                                 2106+                                                                         01-01254
                                 2107+                                                                         01-01255
                                 2108          IODEVICE ADDRESS=(640,2),UNIT=3350,CUNUMBR=002                  00001100
                                 2109+*, 10 IS THE PREVIOUS MVSCP STATEMENT NUMBER                             02-01441
                                 2110+                                                                         02-01442
                                 2152+                                                                         01-01254
                                 2153+                                                                         01-01255
                                 2154          IODEVICE ADDRESS=(820,2),UNIT=3380,FEATURE=AUTOPOLL,           X00001110
                                                     CUNUMBR=041
                                 2155+*, 11 IS THE PREVIOUS MVSCP STATEMENT NUMBER                             02-01441
                                 2156+                                                                         02-01442
                                 2200+                                                                         01-01254
                                 2201+                                                                         01-01255
                                 2202          IODEVICE ADDRESS=(A40,4),UNIT=3380,CUNUMBR=047                  00001110
                                 2203+*, 12 IS THE PREVIOUS MVSCP STATEMENT NUMBER                             02-01441
                                 2204+                                                                         02-01442
                                 2246+                                                                         01-01254
                                 2247+                                                                         01-01255
```

Figure 10-2. Assembler Listing Example

CBP100I **VALIDATE PROCESSING ASSUMED. INVALID INPUT PARAMETER STRING:** {*string*|**NONE SPECIFIED**}

Explanation: The input parameter string specified on the EXEC statement of the JCL was invalid, or no input string was specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the input parameter string on the EXEC statement and rerun the job.

CBP101I **ESTAE ENTERED: SYSTEM CODE** *system-code* [**REASON CODE** *reason-code*]

Explanation: An unexpected error occurred, causing the ESTAE to get control. The reason code appears only when the system code is X'088'.

System Action: MVS configuration program execution terminates.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP102I **CONFIGURATION ID NOT SPECIFIED**

Explanation: The MVS configuration program input stream does not contain an IOCONFIG statement that specifies the configuration identifier.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add a correctly coded IOCONFIG statement to the input stream and rerun the job.

CBP103I **MORE THAN ONE CONFIGURATION ID SPECIFIED**

Explanation: The MVS configuration program input stream contains more than one IOCONFIG statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration output, however.

System Programmer Response: Remove the duplicate statement from the input stream and rerun the job.

CBP104I **DIAGNOSTIC ENTRY NOT ON STACK**

Explanation: A module invoked by the MVS configuration program made an invalid attempt to remove a diagnostic entry from the stack.

System Action: MVS configuration program execution terminates with system ABEND '088'X, reason code '104'X. The correctness of the data in the variable recording area (VRA) of the system diagnostic work area (SDWA) cannot be guaranteed.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP105I **INPUT DDNAME LIST IS NOT VALID, {DDNAME LIST TOO LONG | NULL DDNAME OVERRIDDEN}**

Explanation: The MVS configuration program was invoked with a DDNAME list that was invalid. Either too many DDNAMES were provided, or a null DDNAME was overridden.

System Action: MVS configuration program execution terminates with system ABEND '088'X and reason code '105'X.

System Programmer Response: Correct the DDNAME list and rerun the job. For a description of the DDNAME list, see "Standard DD Name List" on page 8-19.

CBP106I **DUPLICATE ATTEMPT TO ADD DIAGNOSTIC STACK ENTRY**

Explanation: A module invoked by the MVS configuration program attempted to add an entry to the diagnostic stack while the entry was already on the stack.

System Action: MVS configuration program execution terminates with system ABEND '088'X and reason code '106'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP110I **OPEN FAILED FOR DDNAME** *ddname*

Explanation: The MVS configuration program was unable to open the file associated with the specified DDNAME.

System Action: MVS configuration program execution terminates.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP111I **FILE CODE** *file-code* **AND/OR FUNCTION CODE** *function-code* **INVALID**

Explanation: An MVS configuration program routine specified an invalid file code.

System Action: MVS configuration program execution terminates with a system ABEND code of '088'X and a reason code of '111'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP112I **MESSAGE ISSUED WHILE THE MESSAGE LOG FILE WAS CLOSED**

Explanation: An MVS configuration program routine attempted to issue a message while the message log file was closed.

System Action: MVS configuration program execution terminates with a system ABEND code of '088'and a reason code of '112'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP115I **RECORD LENGTH FOR** *ddname* **FILE MUST BE** *record-length*

Explanation: An attempt has been made to open the *ddname* file with an incorrect record length. *ddname* is the DDNAME of the file. *record-length* is the valid value or range for the record length.

System Action: MVS configuration program execution terminates.

System Programmer Response: Correct the JCL and rerun the job.

CBP120I **ASSEMBLER {RETURN|COMPLETION} CODE WAS** *code*

Explanation: The assembler encountered an error while parsing the input statements. *code* is the RETURN code or COMPLETION code.

System Action: MVS configuration program execution terminates.

System Programmer Response: This problem is probably caused by either:

- An input stream error.
- Insufficient space on an assembler work file.
- An END statement was included in the input stream.

Assembler messages, if there are any, are on the SYSASMP data set.

After fixing the problem, rerun the job.

CBP121I **NO RECORDS IN MVSCP INPUT FILE**

Explanation: The MVS configuration program input data set is empty.

System Action: MVS configuration program execution terminates.

System Programmer Response: Verify that you specified the correct data set name. After correcting the problem, rerun the job.

CBP140I **ACON LENGTH IS NOT THREE OR FOUR**

Explanation: Logic error within the MVS configuration program. Only three or four byte ACONs can be resolved.

System Action: MVS configuration program execution terminates with a system ABEND code of '088'X and a reason code of '140'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP.*

CBP141I **ACON POSITION POINTER IS NOT WITHIN ITS OWN TABLE/CSECT**

Explanation: Each ACON must be defined within its own table.

System Action: MVS configuration program execution terminates with a system ABEND code of '088'X and a reason code of '141'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP.*

CBP142I **ACON RELOCATION POINTER TARGET IS NOT WITHIN ITS OWN MODULE**

Explanation: The value at the ACON must be an address that is defined by one of the tables of the module.

System Action: MVS configuration program execution terminates with a system ABEND code of '088'X and a reason code of '142'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP.*

CBP143I **I/O CONFIGURATION MEMBERS ON** *ddname* **DATA SET ARE INVALID. LINK EDIT {RETURN CODE | COMP CODE} WAS** *code*

Explanation: The linkage editor was unable to successfully write the I/O configuration members to the *ddname* data set. (*ddname* is SYSLMOD, unless overridden in the MVS configuration program input DDNAME list). The linkage editor return code or completion code indicates the reason for failure. Only I/O configuration members that were being written during this invocation of the MVS configuration program are in error.

System Action: None.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP144I **OUTPUT TO** *ddname* **NOT REQUESTED. MODE IS {DEBUG | VALIDATE}**

Explanation: The I/O configuration members were not written to the *ddname* data set. (*ddname* is SYSLMOD, unless overridden in the MVS configuration program input DDNAME list). The members were not written because DEBUG or VALIDATE mode was selected.

System Action: Normal completion.

System Programmer Response: If you wish to have the I/O configuration members written, select the appropriate mode (IODEF or EDT) and resubmit the job.

CBP145I **NO OUTPUT WRITTEN TO** *ddname* **DATA SET. VALIDATE PROCESSING FORCED DUE TO ERROR.**

Explanation: An error prevented the MVS configuration program from writing the I/O configuration members to the *ddname* data set. *ddname* is SYSLMOD, unless overridden in the MVS configuration program input DDNAME list.

System Action: No I/O configuration members were written.

System Programmer Response: Correct the errors indicated by other messages and rerun the job.

CBP146I **SECOND REQUEST TO QUEUE THE SAME CSECT WITHIN THE SAME MODULE**

Explanation: A second CALL to the relocation table list builder was made to queue an RTL that was previously queued (same module and CSECT names), or the caller modified restricted RTL field RTLQUED.

System Action: MVS configuration program execution terminates with a system ABEND of '088'X and a reason code of '146'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP147I **ATTACH PROCESSING FAILED. RETURN CODE WAS** *return-code*

Explanation: The MVS configuration program attempted to ATTACH the linkage editor. The ATTACH failed.

System Action: MVS configuration program execution terminates with a system ABEND of '088'X and a reason code of '147'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP148I **LRECL MUST BE 80 FOR DDNAME** *ddname*

Explanation: The LRECL parameter on the *ddname* DD statement specifies a value other than 80.

System Action: MVS configuration program execution terminates.

System Programmer Response: Change LRECL to 80 and rerun the job.

CBP149I **I/O CONFIGURATION MEMBERS SUCCESSFULLY WRITTEN TO *ddname* DATA SET**

Explanation: The linkage editor has written the I/O configuration members to the *ddname* data set. (*ddname* is SYSLMOD, unless overridden in the MVS configuration program input DDNAME list). The linkage editor return code was zero.

System Action: None

System Programmer Response: None.

CBP200I **UNIT= IGNORED FOR *name* WHICH IS A GENERIC.**

Explanation: The UNIT= parameter was specified on a UNITNAME statement whose NAME= parameter specified a generic name. When the NAME= parameter specifies a generic name, the UNIT= parameter must be omitted.

System Action: Processing continues normally. The MVS configuration program ignores the UNIT= parameter, however.

System Programmer Response: Remove the UNIT= parameter from the UNITNAME statement or change the NAME= parameter to specify a non-generic name.

CBP201I **VIO=YES MUST BE SPECIFIED WITH GENERIC *generic-name*.**

Explanation: A generic name was specified on the NAME= parameter of the UNITNAME statement without specifying VIO=YES.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add VIO=YES to the statement, of if message CBP200I was also issued, see the **System Programmer Response** for that message. After correcting the problem rerun the job.

CBP202I **UNITNAME STATEMENT IGNORED. *name* IS A RESERVED NAME.**

Explanation: The NAME= parameter of a UNITNAME statement specified an IBM-reserved name. Currently, the only reserved name is SYSALLDA.

System Action: Processing continues normally. The MVS configuration program ignores the UNITNAME statement, however.

System Programmer Response: Select another name or remove the statement from the input stream.

CBP203I **UNITNAME STATEMENT FOR *name* IGNORED. DUPLICATE ADDRESS *device number*.**

Explanation: A device number was specified more than once on the UNIT= parameter of one or more statements for the given *name*. Either the device number was coded more than once or the replication factor (shown as *n* on the UNIT= parameter of the UNITNAME statement) caused the device number to be specified more than once.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Remove the duplicate device number and rerun the job.

CBP204I **VIO=YES WAS SPECIFIED FOR *name* WHICH IS NOT A DIRECT ACCESS DEVICE.**

Explanation: The generic name of a device that is not a direct access device was specified with VIO=YES.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the UNITNAME statement that is in error and rerun the job. Possible errors are:

- The NAME parameter value is misspelled
- A generic name of a non-DASD device was specified

CBP205I **UNITNAME STATEMENT IGNORED. *name* WAS PREVIOUSLY DEFINED.**

Explanation: The indicated name has been previously defined on a UNITNAME statement that is not contiguous to this UNITNAME statement.

System Action: Processing continues normally. The MVS configuration program ignores this UNITNAME statement, however.

System Programmer Response: Either change the NAME= parameter on this UNITNAME statement, or move this UNITNAME statement so it is contiguous with other UNITNAME statements that specify the same NAME= parameter value.

CBP206I **NO UNIT PARAMETER SPECIFIED FOR name.**

Explanation: On a UNITNAME statement, the NAME= parameter specified a non-generic name, but the UNIT= parameter was omitted from the statement. When a non-generic name is specified, the UNIT= parameter is required.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the NAME= parameter to specify a valid generic name or code the UNIT= parameter. Rerun the job.

CBP207I **VIO SPECIFICATION FOR *name* INCONSISTENT, VIO={YES | NO}.**

Explanation: The same name (NAME=) was specified on multiple UNITNAME statements, but the VIO= parameter was not the same on all statements. All of the UNITNAME statements that specify *name* are printed.

System Action: Processing continues. The MVS configuration program assumes that you wanted to code VIO=YES on all of the statements. The named group of devices is considered VIO eligible if one of the generics corresponding to the name is a direct access device.

System Programmer Response: Change the indicated UNITNAME statements so the VIO= parameter is consistent on all of the statements.

CBP208I **FIRST EDT STATEMENT MUST PRECEDE UNITNAME STATEMENTS.**

Explanation: In the input stream, a UNITNAME statement appears before the first EDT statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: If the input stream contains only one EDT statement, place it ahead of all UNITNAME statements. If the input stream contains multiple EDT statements, place each EDT statement immediately ahead of its corresponding UNITNAME statements. Rerun the job.

CBP209I **UNIT** *device-number* **SPECIFIED FOR** *name* **IS NOT A VALID DEVICE.**

Explanation: The UNIT= parameter of a UNITNAME statement specifies an unrecognized device number. Either the device number is not defined on an IODEVICE statement or the device number is a non-base exposure of a multiple exposure device. If the device number is a non-base exposure, you will also receive message CBP219I.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: If the device was not defined, define the device on an IODEVICE statement. If the device number is a non-base exposure of a multiple exposure device, change it to a base exposure. Rerun the job.

CBP210I *id* **SPECIFIED OR DEFAULTED AS THE ID FOR MORE THAN ONE EDT.**

Explanation: Multiple EDTs were input with two specifying the same identifier.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Check the ID= parameter on the EDT statements. If there are duplicate identifiers specified, change one of them and rerun the job.

You may omit the ID= parameter from only one EDT statement. If you have omitted the ID= parameter from more than one statement, correct the problem and rerun the job.

If you have omitted the ID= parameter from only one EDT statement, look for another EDT statement that specifies an identifier that is the same as the I/O configuration identifier. Change the identifier on this EDT statement and rerun the job.

CBP211I **EDT** *id*: **NO VALID DEVICES SPECIFIED FOR** *name*.

Explanation: For EDT *id*, all of the device numbers specified on the UNIT= parameter of a UNITNAME statement were undefined (they were not defined on an IODEVICE statement). The UNITNAME statement in question is the one whose NAME= parameter specifies the name indicated in the message text. In addition to this message, you will also receive message CBP209I for each undefined device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Locate all affected UNITNAME statements (there may be more than one). Either code an IODEVICE statement for each undefined device, or remove the undefined device numbers from the UNITNAME statements. After correcting the problem, rerun the job.

CBP212I **EDT** *id*: **IF VIO = YES, AT LEAST ONE DEVICE SPECIFIED FOR** *name* **MUST BE A DIRECT ACCESS DEVICE.**

Explanation: For EDT *id*, a UNITNAME statement that specified an esoteric name also specified VIO=YES. The UNIT= parameter, however, did not specify any direct access devices.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: For the specified EDT, locate all of the UNITNAME statements that specify the indicated name. Change the UNITNAME statements so the UNIT= parameter specifies at least one direct access device or remove the VIO=YES specification. After correcting the problem, rerun the job.

CBP213I *name* **IN DEVPREF PARAMETER OMITTED FROM PREFTAB NOT A VALID GENERIC NAME.**

Explanation: The name specified in the DEVPREF parameter of the EDT statement is not a defined generic name.

System Action: Processing continues normally. The MVS configuration program, however, omits the name from the device preference table.

System Programmer Response: Check the DEVPREF= parameter. Either the unit information module (UIM) that recognizes the name is missing, or the name is misspelled. Correct the problem and rerun the job.

CBP214I *name* **IS OMITTED FROM PREFTAB. NO DEVICE DEFINED IN I/O CONFIGURATION.**

Explanation: There was no device defined in the I/O configuration for the name specified on the DEVPREF Keyword.

System Action: Processing continues normally. The MVS configuration program, however, omits the name from the device preference table.

System Programmer Response: Either the IODEVICE statement for the device is missing, or the name is misspelled. Correct the problem and rerun the job.

CBP215I **EDT** *xx*: **VIO=YES ASSUMED FOR** *name*.

Explanation: For EDT *id*, the input stream contains several UNITNAME statements that specify the same NAME= parameter. At least one of these statements, but not all of them, also specify VIO=YES.

System Action: Processing continues normally. The MVS configuration program processes the statements as if they all specified VIO=YES.

System Programmer Response: To avoid receiving this message in the future, change the statements so the VIO= parameter is consistent.

CBP216I **UNIT NAME (***generic-name***) NOT A VALID GENERIC NAME. NO DEVICE DEFINED IN THE I/O CONFIGURATION.**

Explanation: There are no devices defined for the generic name specified on a UNITNAME statement.

System Action: The MVS configuration program ignores the UNITNAME statement and continues normal processing.

System Programmer Response: To correct this problem, do one of the following:

- Include an IODEVICE statement that defines a device belonging to the generic name
- Remove the UNITNAME statement from the input stream

CBP217I **DUPLICATE DEVPREF PARAMETER (***generic-name***) IGNORED.**

Explanation: The same generic name appeared more than once on the DEVPREF parameter of the EDT statement.

System Action: The MVS configuration program treats the name as if it appeared only once. Processing continues normally.

System Programmer Response: To avoid receiving this message in the future, remove the duplicate generic name from the **UNITNAME** statement.

CBP218I **EDT xx: UNIT SPECIFICATION FOR (***esoteric-name***) CAUSES UNIT NUMBER(S) GREATER THAN** *limit*.

Explanation: The device number exceeds the current allowable maximum device number, *limit*. The device number was generated by a replication factor.

System Action: The MVS configuration program performs a syntax check and a semantics check on the rest of the input statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the statement that is in error and rerun the job.

CBP219I **UNIT SPECIFICATION FOR (*esoteric-name*) IGNORED. (*device-number*) IS A NON-ACCESSIBLE DEVICE.**

Explanation: A device number specified on a UNITNAME statement is a non-base exposure of a multiple exposure device. You can specify base exposures only on UNITNAME statements.

System Action: The MVS configuration program omits this device number from the EDT and processing continues normally.

System Programmer Response: If the resulting I/O configuration is acceptable, no action is required. Otherwise, correct the statement that is in error, and rerun the job.

CBP250I **EDT *id*: NOT ALL COMPATIBLE GENERICS GENERATED. INSUFFICIENT GROUP DESCRIPTOR TABLE ENTRIES.**

Explanation: For EDT *id*, there are insufficient entries in the Group Descriptor Table to store information on compatible generics.

System Action: Processing continues normally. However, no more compatible generics are generated.

System Programmer Response: Verify all of the generics in the list.

CBP251I **MORE THAN ONE ATTEMPT BY UIMS TO BUILD A DIT FOR GENERIC DEVICE TYPE value.**

Explanation: The unit information modules (UIMs) made multiple attempts to build a device information table (DIT) for the indicated generic device type.

System Action: MVS configuration program execution terminates with system ABEND '088'X and with reason code '251'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP252I **EDT *id*: DEVICE FEATURES TABLE FOR DEVICE *device* NOT FOUND.**

Explanation: The report routine was attempting to find the device type for device *device*. The device was represented in EDT *id* but did not have a DFT.

System Action: MVS configuration program execution terminates with a system ABEND '088'X and with reason code '252'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP253I **EDT *id*: DEVICE INFORMATION TABLE FOR GENERIC *generic-name* NOT FOUND.**

Explanation: The EDT Build Routine Part One was attempting to find the device information table for the specified generic, but no device information table (DIT) exists.

System Action: MVS configuration program execution terminates with system ABEND '088'X and with reason code '253'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP254I **DUPLICATE GENERIC PREFERENCE FOUND IN UIMS FOR (*generic-name*) AND (*generic-name*).**

Explanation: Two UIMs specified the same generic priority. This message is accompanied by an ABEND code '088'X and a reason code '254'X.

System Action: MVS configuration program processing terminates.

System Programmer Response: If the problem is in an installation-provided UIM, change the generic priority in the UIM to an unused priority, reassemble and relink the UIM, and rerun the job.

If the problem is in an IBM-provided UIM, refer to the diagnostic procedures in the book *Component Diagnosis: MVSCP*.

CBP255I **THE COMPATIBLE-LIST FOR (*name*) CONTAINS DUPLICATE GENERIC DEVICE NAMES.**

Explanation: The named UIM specified a compatible list that contained duplicate devices.

System Action: MVS configuration program execution terminates with system ABEND '088'X and reason code '255'X.

System Programmer Response: If the problem is in an IBM-supplied UIM, contact your IBM representative. Otherwise check the UIM. The list of compatible generic devices contains one of the following:

- A reference to itself
- Two references to the same generic device type.

Remove the duplicate reference, and rerun the job.

CBP330I *device* **DEVICE NOT RECOGNIZED**

Explanation: None of the UIMs recognize the UNIT parameter that is specified on the indicated IODEVICE statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: One of the following actions should correct the problem:

- If you have incorrectly coded the UNIT parameter, either recode the parameter or remove the statement from the input stream. Rerun the job.
- If the UNIT parameter is correctly coded, provide a UIM that recognizes the UNIT parameter. This may require you to write a UIM or to install the product that provides the required UIM. Rerun the job.

CBP331I **DEVICE NUMBER** *device-number* **WAS ALSO SPECIFIED ON STATEMENT nnnnnn**

Explanation: The same device number was specified on both of the indicated IODEVICE statements.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the device number on one of the statements and rerun the job.

CBP332I **NO DIT FOUND FOR GENERIC** *generic* **SPECIFIED IN THE DFP BY UIM** *UIM-name*

Explanation: The named UIM specified a generic in the device features parameter list (DFP) for which no device information table (DIT) was found. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '332'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP3331 **I/O ERROR ENCOUNTERED ON BLDL PROCESSING**

Explanation: An internal MVSCP error occurred while loading UIM modules.

System Action: MVS configuration program execution terminates.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*. After the problem is fixed, rerun the job.

CBP3341 **UIM** *UIM-name* **SPECIFIED INVALID UCB SEGMENT TYPE** *type* **FOR AN ACON POSITION POINTER**

Explanation: The named UIM specified an invalid UCB segment type for an ACON position pointer. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '334'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP3351 **UIM** *UIM-name* **SPECIFIED INVALID UCB SEGMENT TYPE** *type* **FOR AN ACON RELOCATION POINTER**

Explanation: The named UIM specified an invalid UCB segment type for an ACON relocation pointer. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '335'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP3361 **RESIDENT ERP NAMES** *name* **AND** *name* **SPECIFIED FOR TABLE INDEX** *nn*

Explanation: A pair of Resident ERP Table index values match but the associated Resident ERP names do not match.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '336'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP3371 **NO IODEVICE STATEMENTS WERE SPECIFIED**

Explanation: The MVS configuration program processed the input statements but found no IODEVICE statements.

System Action: MVS configuration program execution terminates.

System Programmer Response: Add IODEVICE statements to the input stream, and rerun the job.

CBP338I **UIM *UIM-name* SPECIFIED AN INVALID OFFSET FOR AN ACON POSITION POINTER**

Explanation: The named UIM specified an offset for an ACON position pointer that was not within the specified UCB segment. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '338'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP339I **DCQ ELEMENT NOT FOUND FOR UCB WITH UCBTBYT3 FIELD xx. UIM *uim-name* SPECIFIED THE UCBTBYT3 DATA.**

Explanation: The named UIM specified an invalid device class for a UCB. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '339'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP340I **INVALID DEVICE NUMBER *device-number* SPECIFIED.**

Explanation: The designated IODEVICE statement specified a device number that is greater than 4095.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the device number specification and rerun the job.

CBP341I **PROCESSING TERMINATED BECAUSE THE IIT IS TOO LARGE TO BE LOADED BY IEAIPL40**

Explanation: The IPL information table (IIT) Build Routine built an IIT whose size was greater than 4K.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '341'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP342I **UIM *UIM-name* SPECIFIED INVALID DEVICE NUMBER *device-number* IN THE RELOCATION INFORMATION**

Explanation: The named UIM specified a device number greater than the maximum number allowed. The number was specified in the relocation information for a device. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '342'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP3431 **UIM *UIM-name* SPECIFIED DEVICE NUMBER *device-number* IN THE RELOCATION INFORMATION BUT NO DFT WAS FOUND FOR THIS DEVICE NUMBER**

Explanation: The named UIM specified, in the relocation information, a device number that was not defined on an IODEVICE statement. There is probably a logon error in the named UIM.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '343'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP3441 **THE DCT INFORMATION SPECIFIED FOR DASD TYPE *type* DOES NOT MATCH THE PREVIOUSLY SPECIFIED INFORMATION FOR THIS DASD TYPE**

Explanation: A UIM specified information in the device characteristics parameters list (DCP) which was inconsistent with previously specified information.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '344'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP3451 **SPECIFIED LENGTH OF DCT ENTRY IS GREATER THAN THE MAXIMUM POSSIBLE LENGTH**

Explanation: A UIM specified length for a device characteristics table (DCT) entry was greater than the maximum possible length.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '345'X.

System Programmer Response: Collect any data that will help you isolate the problem to a component or module. If the problem is in IBM-supplied code, contact your IBM representative. If the problem is in an installation-provided UIM and you need information about UIMs, see *System Modifications*. After the problem is fixed, rerun the job.

CBP3461 **INSUFFICIENT SPACE IN DCT FOR CURRENT DCT ENTRY**

Explanation: A UIM attempted to add an entry to the device characteristics table (DCT). Adding that entry would cause the size of the DCT to exceed the maximum possible size.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '346'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP3471 **THE MLT NAME LIST INFORMATION SPECIFIED FOR MLT NAME *name* DOES NOT MATCH THE PREVIOUSLY SPECIFIED INFORMATION FOR THIS MLT NAME**

Explanation: A module list table (MLT) name was specified more than once, at least one time as being optional and at least one time as being required.

System Action: MVS configuration program execution terminates with an ABEND code of '088'X and a reason code of '347'X.

System Programmer Response: Refer to the diagnostic procedures explained in the book *Component Diagnosis: MVSCP*.

CBP348I **UIM *UIM name* SPECIFIED A NUMBER OF MLT NAMES WHICH IS (MORE | LESS) THAN THE (MAXIMUM | MINIMUM) ALLOWED**

Explanation: The named UIM either specified more than five MLT names or specified no MLT names.

System Action: The MVS configuration program terminates execution with an ABEND code of '088'X and a reason code of '348'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP349I **UIM *UIM name* SPECIFIED MORE THAN THE MAXIMUM ALLOWED AMOUNT OF DEVICE DEPENDENT INFORMATION**

Explanation: The named UIM specified more than 256 bytes of device-dependent information. There is probably a logic error in the named UIM.

System Action: The MVS configuration program terminates execution with an ABEND code of '088'X and a reason code of '349'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP350I **REQUIRED PARAMETER *parameter* NOT SPECIFIED FOR *device* DEVICE**

Explanation: A parameter required for the specified device was not specified on the IODEVICE statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the IODEVICE statement and rerun the job.

CBP351I **THE *parameter/feature* PARAMETER/FEATURE IS IGNORED**

Explanation: Either an IODEVICE statement parameter that is not allowed was specified, or an obsolete feature was specified.

System Action: The MVS configuration program ignores the parameter or feature specification and continues to execute in the current mode.

System Programmer Response: To avoid receiving this message in the future, remove the parameter or feature specification from the IODEVICE statement.

CBP352I **FEATURE *feature* NOT RECOGNIZED FOR *device* DEVICE**

Explanation: A unrecognized feature was specified on the IODEVICE statement that defines the indicated device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the IODEVICE statement and rerun the job.

CBP353I **FEATURE** *feature* **WAS SPECIFIED MORE THAN ONCE**

Explanation: A feature was specified more than once on the indicated IODEVICE statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the IODEVICE statement and rerun the job.

CBP354I **UIM** *UIM name* **RECOGNIZES MORE THAN THE MAXIMUM NUMBER OF FEATURES ALLOWED**

Explanation: A UIM indicated that it recognized more than 64 features. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with a system ABEND of '088'X and a reason code of '354'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP355I **UIM** *UIM name* **DID NOT CAUSE ANY DFTS TO BE BUILT FOR A VALID IODEVICE STATEMENT**

Explanation: The named UIM processed an IODEVICE statement, detected no errors on the statement, but failed to build any DFTS. There is probably a logic error in the named UIM.

System Action: MVS configuration program execution terminates with a system ABEND of '088'X and a reason code of '355'X.

System Programmer Response: If the problem is in an installation-provided UIM, correct the problem and rerun the job.

If you suspect that the problem is in an IBM-provided UIM (or in other IBM-provided MVS configuration program code), refer to the book *Component Diagnosis: MVSCP* for diagnostic instructions.

CBP400I **THE** *device* **AND** *device* **DEVICES SHOULD BOTH BE SPECIFIED OR OMITTED FROM THE I/O CONFIGURATION DEFINITION**

Explanation: If both of the devices are not specified in the configuration definition, MVS will be unable to use either device.

System Action: Processing continues in the current mode. This message is issued as a warning.

System Programmer Response: Correct the device specification and rerun the job.

CBP401I *model* **IS NOT A VALID MODEL NUMBER FOR THE** *device* **DEVICE**

Explanation: The specified model number is invalid for the indicated device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the model number specification on the IODEVICE statement and rerun the job.

CBP402I ONLY ONE KEYBOARD MAY BE SPECIFIED

Explanation: The device supports multiple keyboards. Only one may be specified per device, however.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the keyboard specification on the IODEVICE statement and rerun the job.

CBP403I MORE THAN ONE CHARACTER GENERATOR SPECIFIED

Explanation: The device supports multiple character sets. Only one may be specified per device, however.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the character generator specification on the IODEVICE statement and rerun the job.

CBP404I MORE THAN ONE KEY TYPE SPECIFIED

Explanation: The device supports multiple key types. Only one may be specified per device, however.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the keyboard type specification on the IODEVICE statement and rerun the job.

CBP405I FEATURES *feature1* **AND** *feature2* **ARE MUTUALLY EXCLUSIVE FOR THE** *device* **DEVICE**

Explanation: These features cannot be specified together for the device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Delete one of the feature specifications from the IODEVICE statement and rerun the job.

CBP407I ADDRESS *device-number* **SPECIFIED FOR THE 3350P. THE NEXT TO LOW ORDER DIGIT MUST BE EVEN.**

Explanation: The next-to-low order digit (that is the middle digit) of the 3-digit device number must be even (0, 2, 4, 6, 8, A, C, or E).

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the device number specification on the IODEVICE statement and rerun the job.

CBP408I **ADDRESS** *device-number* **SPECIFIED FOR THE** *device*. **THE SUM OF THE LOW ORDER DIGIT PLUS THE REPLICATION FACTOR MUST BE LESS THAN** {**NINE** | **FIVE**}

Explanation: The base UCB device-numbers extend beyond the allowable range of device-numbers. For a 3350P the allowable range is 32; for a 3351P the allowable range is 16.

The sum of the low order digit of the device number plus the replication factor (also called the *number-of-units* on the IODEVICE statement) must be:

- Less than nine (9) for a 3350P
- Less than five (5) for a 3351P

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the ADDRESS= parameter on the IODEVICE statement and rerun the job.

CBP409I *parameter* **PARAMETER IS IGNORED FOR THE** *device* **MODEL** *model*

Explanation: The specified parameter is ignored for this particular model of the *device* device.

System Action: Processing continues in the current mode.

System Programmer Response: To avoid receiving this message in the future, remove the specified parameter from the IODEVICE statement.

CBP410I **FEATURE** *feature1* **REQUIRES THAT FEATURES** *feature2* **AND** *feature3* **BE SPECIFIED, OR THAT FEATURES** *feature4* **AND** *feature5* **BE SPECIFIED**

Explanation: The required combination of features was not specified. When feature *feature1* is specified, either *feature2* and *feature3*, or *feature4* and *feature5* must be specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Specify the required combination of features and rerun the job.

CBP411I **FEATURE AUDALRM REQUIRES A KEYBOARD**

Explanation: When AUDALRM feature is specified, a keyboard must also be specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add a keyboard specification to the indicated IODEVICE statement, or remove the audible alarm specification from the statement. Rerun the job.

CBP412I **FEATURE** *feature1* **REQUIRES THAT FEATURES** *feature2* **AND** *feature3* **BE SPECIFIED**

Explanation: When feature *feature1* is specified, the other named features must also be specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add the required feature specifications to the IODEVICE statement, or remove the specification for feature *feature1*. Rerun the job.

CBP413I FEATURE DATACONV REQUIRES THAT FEATURE 7-TRACK BE SPECIFIED FOR THE 3420 MODELS 3, 5 OR 7 TAPE

Explanation: When feature DATACONV is specified, the feature 7-TRACK must also be specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add the 7-track feature to the IODEVICE statement, or remove the DATACONV feature from the statement. Rerun the job.

CBP414I THE DEFAULT REPLICATION FACTOR FOR THE *device* DEVICE IS TWO.

Explanation: The replication factor (also called the *number-of-units* on the IODEVICE statement) was omitted. For the indicated device, the replication factor defaults to two.

System Action: Processing continues in the current mode.

System Programmer Response: None.

CBP416I *value* IS NOT VALID FOR THE *parameter* PARAMETER FOR THE *device* DEVICE

Explanation: An invalid parameter value was specified for the indicated device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the IODEVICE statement and rerun the job.

CBP417I *parameter1* PARAMETER REQUIRES *parameter2* PARAMETER FOR THE *device* DEVICE

Explanation: When the *parameter1* parameter is specified, the *parameter2* parameter must also be specified for the indicated device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Either add the *parameter2* parameter to the IODEVICE statement, or remove the *parameter1* parameter. Rerun the job.

CBP418I *value* VALUE IS NOT VALID FOR THE *parameter1* PARAMETER WHEN THE *parameter2* PARAMETER HAS VALUE OF *value*

Explanation: Incompatible values have been specified for two different parameters.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the parameter values on the IODEVICE statement so they do not conflict and rerun the job.

CBP419I DEVICE NUMBER DOES NOT END IN ZERO

Explanation: The device number for the device defined on the indicated IODEVICE statement does not have a low-order digit of 0.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the device number on the IODEVICE statement so it has a low-order digit of 0 and rerun the job.

CBP420I **FEATURE** *feature1* **CANNOT BE SPECIFIED WITH EITHER FEATURES** *feature2*, *feature3*, **OR** *feature4*

Explanation: Mutually exclusive features have been specified on an IODEVICE statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the feature specification on the IODEVICE statement and rerun the job.

CBP421I **FEATURE** *feature1* **REQUIRES THAT FEATURE** *feature2* **IS SPECIFIED**

Explanation: Feature *feature1* cannot be specified unless you also specify *feature2*.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the IODEVICE statement to include the required feature, or else delete *feature1*. Then rerun the job.

CBP422I **FEATURE** *feature1* **CANNOT BE SPECIFIED WITH EITHER FEATURES** *feature2* **OR** *feature3*

Explanation: Mutually exclusive features have been specified on the IODEVICE statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Correct the IODEVICE statement to specify a valid combination of features and rerun the job.

CBP423I **FEATURE** *feature* **IS NOT VALID WITH A TCU VALUE OF** *value*

Explanation: The indicated IODEVICE statement specifies a feature and a TCU value that are mutually exclusive.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Remove the feature specification or change the value of the TCU and rerun the job.

CBP424I **MORE GUARANTEED BUFFERS REQUESTED THAN ARE SUPPORTED IN THE DEVICE BUFFER TABLE FOR THE PCU** *pcu-number*

Explanation: The NUMSECT= parameter on the IODEVICE statements for 2250s has requested more guaranteed buffer sections than are available in that table. The affected PCU is *pcu-number*.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the IODEVICE statements for the 2250s to reduce the amount of request buffer space and rerun the job.

CBP426I **MORE THAN FOUR DEVICES REQUESTED PCU** *id*

Explanation: More than four IODEVICE statements for 2250s have specified the same physical control unit. Each physical control unit can support no more than four 2250s.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the PCU parameter on the IODEVICE statements for the 2250s so that no more than four 2250s specify the same physical control unit. Rerun the job.

CBP427I **THE MINIMUM REPLICATION FACTOR FOR THE** *device* **DEVICE IS TWO.**

Explanation: The replication factor (also called the *number-of-units* on the IODEVICE statement) was either omitted or was assigned a value of one. For this device, the replication factor must be at least two.

System Action: The MVS configuration program defaults the replication factor to two and processing continues in the current mode.

System Programmer Response: To avoid receiving this message in the future, change the indicated IODEVICE statement to specify the correct replication factor.

CBP428I **FEATURE PARAMETER IS IGNORED FOR THE BSC1 DEVICE UNLESS TCU = 2701 IS SPECIFIED.**

Explanation: The feature parameter was specified on the indicated IODEVICE statement without TCU = 2701 being specified.

System Action: Processing continues in the current mode.

System Programmer Response: To avoid receiving this message in the future, remove the FEATURE parameter from the indicated IODEVICE statement or specify a different TCU value.

CBP429I **REPLICATION FACTOR OF ONE IS USED. EIGHT DEVICE NUMBERS DEFINED.**

Explanation: The ADDRESS parameter on the indicated IODEVICE statement specified a replication factor (also called *number-of-units* on the IODEVICE statement) greater than one. This device requires a replication factor of one which causes the MVS configuration program to define eight device numbers.

System Action: The MVS configuration program defaults the replication factor to one. Processing continues in the current mode.

System Programmer Response: To avoid receiving this message in the future, change the replication factor to one on the indicated IODEVICE statement.

CBP430I *device1* **AND** *device2* **CANNOT BOTH BE DEFINED FOR THE SAME CONFIGURATION.**

Explanation: The specified devices are mutually exclusive and cannot be defined in the same configuration.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Remove the IODEVICE statement for one of the devices and rerun the job.

CBP431I CHECKING FEATURE REQUIRED WITH 2740C AND 2740X

Explanation: Checking is a required feature for the specified devices.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Include the CHECKING feature on the IODEVICE statements for these devices and rerun the job.

CBP432I *parameter1* **PARAMETER IS IGNORED FOR THE** *device* **WHEN THE** *parameter2* **PARAMETER IS SPECIFIED.**

Explanation: Incompatible parameters, *parameter1* and *parameter2*, were specified on the IODEVICE statement for the named device.

System Action: The MVS configuration program uses *parameter2* instead of *parameter1* and continues processing in the current mode.

System Programmer Response:

- If you want to use *parameter2*, no action is required.
- If you want to use *parameter2* and wish to avoid receiving this message in the future, remove *parameter1* from the IODEVICE statement.
- If you want to use *parameter1*, remove *parameter2* from the IODEVICE statement and then rerun the job.

CBP433I *parameter1* **IS IGNORED FOR THE** *device* **WHEN THE** *parameter2* **PARAMETER IS NOT SPECIFIED.**

Explanation: A parameter, *parameter1*, was coded on an IODEVICE statement, however, a required companion parameter, *parameter2*, was not coded on the IODEVICE statement.

System Action: The MVS configuration program ignores *parameter1* and continues processing in the current mode.

System Programmer Response: Either remove *parameter1* from the IODEVICE statement, or code both *parameter1* and *parameter2* on the IODEVICE statement. If you code both parameters, also rerun the job.

CBP434I FEATURE *feature1* **REQUIRES THAT FEATURES** *feature2* **OR** *feature3* **BE SPECIFIED**

Explanation: When *feature1* is specified, *feature2*, or *feature3*, or both must also be specified.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Add the required feature specifications to the IODEVICE statement, or remove the specification for feature *feature1*. Rerun the job.

CBP500I NO NIPCON STATEMENTS WERE SPECIFIED

Explanation: The MVS configuration program input stream must include at least one NIPCON statement.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Include a NIPCON statement in the input stream and rerun the job.

CBP501I **NIPCON STATEMENT SPECIFIED UNDEFINED DEVICE NUMBER** *device-number*

Explanation: The indicated device number was specified as a NIP console device, but the input stream contained no IODEVICE statement for the device.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Either remove the device number from the NIPCON statement or include an IODEVICE statement for the device. Rerun the job.

CBP502I **DEVICE NUMBER** *device-number* **CANNOT BE USED AS A NIP CONSOLE DEVICE. THIS DEVICE WAS DEFINED BY STATEMENT** *nnnnnn*

Explanation: The indicated device number was specified as a NIP console device. MVS does not support that device as a NIP console, however.

System Action: The MVS configuration program performs a syntax check and a semantics check on the input stream statements. The MVS configuration program does not write the I/O configuration to the output data set, however.

System Programmer Response: Change the NIPCON statement or the IODEVICE statement to specify a device that MVS supports as a NIP console, and rerun the job. For a list of devices that you can define as NIP consoles, see "Defining NIP Consoles" on page 7-9.

CBP503I **DEVICE NUMBER** *device-number* **WAS PREVIOUSLY SPECIFIED AS A NIP CONSOLE DEVICE ON STATEMENT** *nnnnnn*. **THE DUPLICATE SPECIFICATION IS IGNORED.**

Explanation: The indicated device number was previously specified as a NIP console device.

System Action: The first NIP console definition statement for this device is used. Others are ignored.

System Programmer Response: To avoid receiving this message in the future, remove the duplicate NIPCON definition.

CBP600I **JES3 PROCESSING OF EDT** *id* **COMPLETE.**

Explanation: JES3 data for the indicated EDT as well as for the I/O configuration has been successfully generated.

System Action: Processing continues normally.

System Programmer Response: None.

CBP900 to CBP999

Explanation: These messages are reserved for installations to use when they write their own UIMs. Therefore, they are not documented in IBM publications.

Appendix A. Generic Device Types

The MVS configuration program logically groups devices with similar allocation characteristics into groups called generic device types. The MVS configuration program then assigns each group a name called the groups "generic name." Certain parameter values that you code on an IODEVICE statement determines the generic name assigned to the corresponding device. To determine which generic name to assign, the MVS configuration program uses the UNIT parameter and in some cases the MODEL and FEATURE parameters. Other parameters can appear on the IODEVICE statement but are not used to determine the generic name.

This appendix contains a set of tables that identifies generic names. For each generic name the tables also identify the IODEVICE parameter values that cause a device to be assigned the corresponding generic name. Each table contains:

- In the left column -- a generic name. When a table contains more than one name, the names are arranged alphabetically.
- In the right column -- the IODEVICE statement parameter values that cause the MVS configuration program to assign a device the corresponding generic name.

The tables are arranged in ascending alphabetical order according to their headings (Array Processor, Channel-Channel Adapters, etc.).

Array Processor

This generic name	includes only those devices specified as:
3838	UNIT=3838

Channel-Channel Adapters

This generic name	includes only those devices specified as:
CTC	UNIT=CTC

Cryptographic Unit

This generic name	includes only those devices specified as:
3848	UNIT=3848

Direct Access

This generic name	includes only those devices specified as:
3330	UNIT=3330,MODEL=1
3330-1	UNIT=3330,MODEL=11
3340	UNIT=3340
3350	UNIT=3350 UNIT=3350P UNIT=3351P
3375	UNIT=3375
3380	UNIT=3380

Diskette

This generic name	includes only those devices specified as:
3540	UNIT=3540

Drum

This generic name	includes only those devices specified as:
2305-2	UNIT=2305,MODEL=2

Dummy Device

This generic name	includes only those devices specified as:
DUMMY	UNIT=DUMMY

Graphic (display) Devices

This generic name	includes only those devices specified as:
HFGD	UNIT=HFGD
2250-3	UNIT=2250,MODEL=3
3277-1	UNIT=3277,MODEL=1 UNIT=3278,MODEL=1
3277-2	UNIT=3270,MODEL=X UNIT=3277,MODEL=2 UNIT=3278,MODEL=2 UNIT=3278,MODEL=2A UNIT=3278,MODEL=3 UNIT=3278,MODEL=4 UNIT=3279,MODEL=2A UNIT=3279,MODEL=2B UNIT=3279,MODEL=2C UNIT=3279,MODEL=3A UNIT=3279,MODEL=3B
3284-1	UNIT=3284,MODEL=1
3284-2	UNIT=3284,MODEL=2
3286-1	UNIT=3286,MODEL=1
3286-2	UNIT=3286,MODEL=2

Magnetic Tape

This generic name	includes only those devices specified as:
2400	UNIT=2400
2400-1	UNIT=2400,MODEL=1
2400-2	UNIT=2400,MODEL=2
2400-3	UNIT=2400,MODEL=3
2400-4	UNIT=2400,MODEL=4
3400-2	UNIT=3420,Model=3,FEATURE=(7-TRACK,DATACONV) UNIT=3420,Model=5,FEATURE=(7-TRACK,DATACONV) UNIT=3420,Model=7,FEATURE=(7-TRACK,DATACONV)
3400-3	UNIT=3420,Model=3,FEATURE=9-TRACK UNIT=3420,Model=5,FEATURE=9-TRACK UNIT=3420,Model=7,FEATURE=9-TRACK
3400-4	UNIT=3420,Model=3,FEATURE=(9-TRACK,DUALDENS) UNIT=3420,Model=5,FEATURE=(9-TRACK,DUALDENS) UNIT=3420,Model=7,FEATURE=(9-TRACK,DUALDENS)
3400-5	UNIT=3420,Model=4,FEATURE=9-TRACK UNIT=3420,Model=6,FEATURE=9-TRACK UNIT=3420,MODEL=8,FEATURE=9-TRACK
3400-6	UNIT=3420,Model=4,FEATURE=(9-TRACK,OPT1600) UNIT=3420,Model=6,FEATURE=(9-TRACK,OPT1600) UNIT=3420,Model=8,FEATURE=(9-TRACK,OPT1600) UNIT=3422,FEATURE=(9-TRACK,OPT1600) UNIT=3430,FEATURE=(9-TRACK,OPT1600)
3400-9	UNIT=3420C
3480	UNIT=3480

Mass Storage Subsystem

This generic name	includes only those devices specified as:
3330V	UNIT=3330V
3851	UNIT=3851

Optical and Magnetic Character Readers

This generic name	includes only those devices specified as:
1287	UNIT=1287
1288	UNIT=1288
3886	UNIT=3886
3890	UNIT=3890
3895	UNIT=3895

Printers

This generic name	includes only those devices specified as:
1403	UNIT=1403,MODEL=N1 UNIT=1403,MODEL=2 UNIT=1403,MODEL=7
3203	UNIT=3203,MODEL=4 UNIT=3203,MODEL=5
3211	UNIT=3211
3800	UNIT=3800,MODEL=1 UNIT=3800,MODEL=3 UNIT=3800,Model=8
4245	UNIT=4245
4248	UNIT=4248

Readers and Punches

This generic name	includes only those devices specified as:
2501	UNIT=2501,MODEL=B1 UNIT=2501,MODEL=B2
2540	UNIT=2540R
2540-2	UNIT=2540P
3505	UNIT=3505
3525	UNIT=3525

Telecommunication Devices

This generic name	includes only those devices specified as:
AAA1	UNIT=1030
AAA2	UNIT=1050
AAA5	UNIT=115A
AAA6	UNIT=83B3
AAA7	UNIT=TWX
AAA8	UNIT=2740
AAA9	UNIT=BSC1
AAAA	UNIT=BSC2
AAAB	UNIT=BSC3
AAAC	UNIT=7770
AAAD	UNIT=WTTA
AAAE	UNIT=2741P
AAAF	UNIT=2741C
AAAG	UNIT=1050X
AAAH	UNIT=2740X
AAAI	UNIT=2740C
3704	UNIT=3704
3705	UNIT=3705
3791L	UNIT=3791L

Appendix B. IBM Provided Device Preference Table

The following list shows the device order that MVS uses when it attempts an allocation to satisfy a request for a device from an esoteric device group. The order of the IBM-defined list ensures that MVS always tries to allocate the fastest possible available device.

For each EDT that you define, you may change the order of the list. Code the changes on the DEVPREF= parameter of the EDT statement.

Device Type	Generic Name
drum	2305-2
direct access	3380 3350 3375 3330-1 3330 3340
magnetic tape	3480 3400-9 3400-5 3400-3 3400-6 3400-4 3400-2 2400-3 2400 2400-4 2400-2 2400-1
printers	3800 4248[1] 4245 3211 3203 1403
readers/punches	2501 3505 3525 2540 2540-2

1 Because the 3262 Model 5 is generated as a 4248 printer, the entry 4248 in the installation device preference table may refer to a 4248 printer, a 3262 Model 5 printer, or both. If a 3262 Model 5 printer is installed on your system, and the 4248 entry in your installation device preference table precedes the entries for the 4245 or the 3800 printers, your system may allocate the slower 3262 Model 5 printer in preference to the faster 4245, 4248, or 3800.

graphic (display) devices	HFGD 2250-3 3277-1 3277-2 3284-1 3284-2 3286-1 3286-2
optical or magnetic character readers	3890 3886 1287 1288 3895
diskette	3540
mass storage subsystem	3851 3330V
array processor	3838
telecommunication device	AAA1 AAA2 AAA5 AAA6 AAA7 AAA8 AAA9 AAAA AAAB AAAC AAAD AAAE AAAF 3705 3791L 3704
channel-to-channel CU	CTC
telecommunication devices	AAAG AAAH AAAI
miscellaneous devices	DUMMY 3848

Appendix C. Determining 3350P and 3351P Device Number Assignments

The values that you code on the ADDRESS= parameter of the 3350P or 3351P IODEVICE statement determines:

- How many device numbers the MVS configuration program assigns to the device
- The values of the assigned device numbers

3350P Device Number Assignments

The ADDRESS= parameter on the IODEVICE statement that defines a 3350P has one of the following general forms:

```
IODEVICE UNIT=3350P,ADDRESS=device-number,...
```

```
IODEVICE UNIT=3350P,ADDRESS=(device-number,number-of-units),...
```

Use the following chart to calculate the value of the device numbers that the MVS configuration program assigns to a 3350P. The variable *n*, shown in the chart, represents the value of the device number that you specify on the ADDRESS= parameter. For example, if you code ADDRESS=(200,2), substitute 200 for *n*.

If the value you specify for *number-of-units* is:

- 1 or 2 or is omitted, The MVS configuration program assigns the first two rows of device numbers shown in the chart.
- 3, The MVS configuration program assigns the top three rows of device numbers shown in the chart.
- 4, The MVS configuration program assigns the top four rows of device numbers shown in the chart.
- 5, The MVS configuration program assigns the top five rows of device numbers shown in the chart.
- 6, The MVS configuration program assigns the top six rows of device numbers shown in the chart.
- 7, The MVS configuration program assigns the top seven rows of device numbers shown in the chart.
- 8, The MVS configuration program assigns all of the device numbers shown in the chart.

n	n + '08'X	n + '10'X	n + '18'X
(n+1)	(n+1) + '08'X	(n+1) + '10'X	(n+1) + '18'X
(n+2)	(n+2) + '08'X	(n+2) + '10'X	(n+2) + '18'X
(n+3)	(n+3) + '08'X	(n+3) + '10'X	(n+3) + '18'X
(n+4)	(n+4) + '08'X	(n+4) + '10'X	(n+4) + '18'X
(n+5)	(n+5) + '08'X	(n+5) + '10'X	(n+5) + '18'X
(n+6)	(n+6) + '08'X	(n+6) + '10'X	(n+6) + '18'X
(n+7)	(n+7) + '08'X	(n+7) + '10'X	(n+7) + '18'X

3350P Base Exposures

The device numbers shown in the leftmost column (n, n+1, n+2, etc.) are called the 3350P's *base exposures*. If you include 3350P device numbers on a UNITNAME statement, you may include the base exposures only. The other device numbers are called the 3350P's non-base exposures.

3350P Examples

If you code ADDRESS=000, or ADDRESS=(000,1), or ADDRESS=(000,2), the MVS configuration program assigns the device numbers shown below. Base exposures are 000 and 001.

000	008	010	018
001	009	011	019

If you code ADDRESS=(000,8), the MVS configuration program assigns the device numbers shown below. Base exposures are 000, 001, 002, 003, 004, 005, 006, and 007.

000	008	010	018
001	009	011	019
002	00A	012	01A
003	00B	013	01B
004	00C	014	01C
005	00D	015	01D
006	00E	016	01E
007	00F	017	01F

3351P Device Number Assignments

The ADDRESS= parameter on the IODEVICE statement that defines a 3351P has one of the following general forms:

```
IODEVICE UNIT=3351P,ADDRESS=device-number,...
```

```
IODEVICE UNIT=3351P,ADDRESS=(device-number,number-of-units),...
```

Use the following chart to calculate the value of the device numbers that the MVS configuration program will assign to a 3351P. The variable *n*, shown in the chart, represents the value of the device number that you specify on the ADDRESS= parameter. For example, if you code ADDRESS=(200,2), substitute 200 for *n*.

If the value you specify for *number-of-units* is:

- 1 or 2 or is omitted, The MVS configuration program assigns the top two rows of device numbers shown in the chart.
- 3, The MVS configuration program assigns the top three rows of device numbers shown in the chart.
- 4, The MVS configuration program assigns all of the device numbers shown in the chart.

n	n + '04'X	n + '08'X	n + '0C'X
(n+1)	(n+1) + '04'X	(n+1) + '08'X	(n+1) + '0C'X
(n+2)	(n+2) + '04'X	(n+2) + '08'X	(n+2) + '0C'X
(n+3)	(n+3) + '04'X	(n+3) + '08'X	(n+3) + '0C'X

3351P Base Exposures

The device numbers shown in the leftmost column (n, n+1, n+2, and n+3) are called the 3351P's *base exposures*. If you include 3351P device numbers on a UNITNAME statement, you may include the base exposures only. The other device numbers are called the 3351P's non-base exposures.

3351P Examples

If you code ADDRESS=000, or ADDRESS=(000,1), or ADDRESS=(000,2), the MVS configuration program assigns the device numbers shown below. Base exposures are 000 and 001.

```
000   004   008   00C
001   005   009   00D
```

If you code ADDRESS=(000,4), the MVS configuration program assigns the following device numbers. Base exposures are 000, 001, 002, and 003.

```
000   004   008   00C
001   005   009   00D
002   006   00A   00E
003   007   00B   00F
```

Appendix D. MVS Configuration Program Example

The following example shows a combined IOCP and MVS configuration program input stream.

```
PS2      IOCONFIG ID=02
*
ID02  ID MSG1='EXAMPLE  CONFIGURATION '
*
         CHPID PATH=00,TYPE=BY
         CHPID PATH=((01),(02),(03),(04),(05),(06),(07),(08)),TYPE=BL
         CHPID PATH=((09),(0A),(0B),(0C),(0D),(0E),(0F),(10)),TYPE=BL
         CHPID PATH=((11),(12),(13),(14),(15),(16),(17),(18)),TYPE=BL
         CHPID PATH=((19),(1A),(1B),(1C),(1D),(1E),(1F),(20)),TYPE=BL
         CHPID PATH=((21),(22),(23),(24),(25),(26),(27),(28)),TYPE=BL
         CHPID PATH=((29),(2A),(2B),(2C),(2D),(2E),(2F)),TYPE=BL
*
*
********************* UNIT ADDRESSES 000-0FF ************************
*
         CNTLUNIT CUNUMBR=00,UNIT=3811,UNITADD=((03,4)),SHARED=N,      *
               PATH=(00)
         IODEVICE  UNIT=3211,ADDRESS=(004,3),CUNUMBR=00
         IODEVICE  UNIT=DUMMY,ADDRESS=003,CUNUMBR=00
*IOCP
         CNTLUNIT CUNUMBR=02,UNIT=3203,UNITADD=((07,1)),SHARED=N,      *
               PATH=(00)
         IODEVICE UNIT=3203,MODEL=4,ADDRESS=(007),UNITADD=07,CUNUMBR=02
*IOCP
         CNTLUNIT CUNUMBR=03,UNIT=3800,UNITADD=((0C,1)),SHARED=N,      *
               PATH=(00)
         IODEVICE  UNIT=3800,FEATURE=CGS2,                             *
               ADDRESS=(00C,1),UNITADD=0C,CUNUMBR=03
*IOCP
         CNTLUNIT CUNUMBR=04,UNIT=3505,UNITADD=((12,4)),SHARED=N,      *
               PATH=(00)
         IODEVICE  UNIT=3505,ADDRESS=(012,1),CUNUMBR=04
         IODEVICE  UNIT=3525,ADDRESS=(013,1),CUNUMBR=04
         IODEVICE  UNIT=1288,ADDRESS=(014,1),CUNUMBR=04
*IOCP
         CNTLUNIT CUNUMBR=05,UNIT=3880,UNITADD=((80,16)),SHARED=N,     *
               PATH=(28,2F),PROTOCL=S
         IODEVICE  UNIT=3351P,ADDRESS=(080,2),CUNUMBR=05,UNITADD=80
*IOCP
         CNTLUNIT CUNUMBR=06,UNIT=3880,UNITADD=((C0,32)),SHARED=N,     *
               PATH=(20,2A),PROTOCL=S
         IODEVICE  UNIT=3350P,ADDRESS=(0C0,8),CUNUMBR=06,UNITADD=C0
*IOCP
         CNTLUNIT CUNUMBR=07,UNIT=3705,UNITADD=((E0,16),(F1,3)),       *
               PATH=00,SHARED=N
         IODEVICE  UNIT=2741C,ADDRESS=(0E0,16),CUNUMBR=07,TCU=2703,    *
               ADAPTER=IBM1,FEATURE=AUTOANSR
         IODEVICE  UNIT=3705,ADDRESS=(0F1,3),CUNUMBR=07,ADAPTER=CA1
```

```
*
*
********************* UNIT ADDRESSES 100-1FF ************************
*
         CNTLUNIT CUNUMBR=10,UNIT=3880,UNITADD=((00,32)),SHARED=N,     *
               PATH=(01,0B),PROTOCL=S
         IODEVICE  UNIT=3351P,ADDRESS=(100,2),CUNUMBR=10,UNITADD=00
         IODEVICE  UNIT=3351P,ADDRESS=(110,2),CUNUMBR=10,UNITADD=10
*IOCP
         CNTLUNIT CUNUMBR=11,UNIT=3880,UNITADD=((20,32)),SHARED=N,     *
               PATH=(06,1F),PROTOCL=S
         CNTLUNIT CUNUMBR=16,UNIT=3880,UNITADD=((20,32)),SHARED=N,     *
               PATH=(07,29),PROTOCL=S
         IODEVICE  UNIT=3351P,ADDRESS=(120,4),CUNUMBR=(11,16)
         IODEVICE  UNIT=3351P,ADDRESS=(130,4),CUNUMBR=(11,16)
*IOCP
         CNTLUNIT CUNUMBR=13,UNIT=3880,UNITADD=((60,32)),SHARED=N,     *
               PATH=(10,1A),PROTOCL=S
         IODEVICE  UNIT=3350P,ADDRESS=(160,4),UNITADD=60,CUNUMBR=13
*
*
********************* UNIT ADDRESSES 200-2FF ************************
*
         CNTLUNIT  CUNUMBR=20,UNIT=3880,UNITADD=((20,16)),             *
               PATH=(13,2F),PROTOCL=S,SHARED=N
         CNTLUNIT  CUNUMBR=21,UNIT=3880,UNITADD=((20,16)),             *
               PATH=(10,25),PROTOCL=S,SHARED=N
         IODEVICE  UNIT=3380,ADDRESS=(220,16),UNITADD=20,              *
               CUNUMBR=(20,21),STADET=N,FEATURE=ALTCTRL
*IOCP
         CNTLUNIT  CUNUMBR=22,UNIT=3880,UNITADD=((70,16)),             *
               PATH=(1B),PROTOCL=S,SHARED=N
         CNTLUNIT  CUNUMBR=23,UNIT=3880,UNITADD=((70,16)),             *
               PATH=(25,17),PROTOCL=S,SHARED=N
         IODEVICE  UNIT=3380,FEATURE=(ALTCTRL,SHARED),                 *
               ADDRESS=(270,16),UNITADD=70,CUNUMBR=(22,23)
*
*
********************* UNIT ADDRESSES 300-3FF ************************
*
         CNTLUNIT  CUNUMBR=30,UNIT=3274,UNITADD=((60,32)),             *
               PATH=(05),SHARED=N
         IODEVICE  UNIT=3270,MODEL=X,                                  *
               FEATURE=(EBKY3277,KB78KEY,SELPEN,DOCHAR,AUDALRM,PTREAD),*
               ADDRESS=(360,16),UNITADD=60,CUNUMBR=30
*IOCP
         CNTLUNIT  CUNUMBR=31,UNIT=3274,UNITADD=((C0,32)),             *
               PATH=(0F),SHARED=N
         IODEVICE  UNIT=3270,MODEL=X,                                  *
               FEATURE=(EBKY3277,KB78KEY,SELPEN,DOCHAR,AUDALRM),       *
               ADDRESS=(3C0,16),UNITADD=C0,CUNUMBR=31
         IODEVICE  UNIT=3270,MODEL=X,                                  *
               FEATURE=(EBKY3277,KB78KEY,SELPEN,DOCHAR,AUDALRM),       *
               ADDRESS=(3D0,6),UNITADD=D0,CUNUMBR=31
         IODEVICE  UNIT=3286,MODEL=2,                                  *
               FEATURE=DOCHAR,                                         *
               ADDRESS=(3D6,2),UNITADD=D6,CUNUMBR=31
         IODEVICE  UNIT=3270,MODEL=X,                                  *
               FEATURE=(EBKY3277,KB78KEY,SELPEN,DOCHAR,AUDALRM),       *
               ADDRESS=(3D8,8),UNITADD=D8,CUNUMBR=31
```

```
*
*
********************* UNIT ADDRESSES 400-4FF ************************
*
        CNTLUNIT  CUNUMBR=48,UNIT=3830,UNITADD=((10,16)),            *
              PATH=(0A,14),SHARED=N
        IODEVICE  UNIT=3330,ADDRESS=(410,16),FEATURE=SHARED,CUNUMBR=48
*IOCP
        CNTLUNIT  CUNUMBR=49,UNIT=3880,UNITADD=((20,16)),            *
              PATH=(02),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=4A,UNIT=3880,UNITADD=((20,16)),            *
              PATH=(0C,16),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,FEATURE=(ALTCTRL,SHARED),ADDRESS=(420,16),*
              CUNUMBR=(49,4A)
*IOCP
        CNTLUNIT  CUNUMBR=41,UNIT=3880,UNITADD=((70,16)),            *
              PATH=(0A,14),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,FEATURE=(ALTCTRL,SHARED),                *
              ADDRESS=(470,16),UNITADD=70,CUNUMBR=41
*IOCP
        CNTLUNIT  CUNUMBR=42,UNIT=3830,UNITADD=((A0,16)),            *
              PATH=(02,24),SHARED=N
        CNTLUNIT  CUNUMBR=43,UNIT=3830,UNITADD=((A0,16)),            *
              PATH=(0C,16),SHARED=N
        IODEVICE  UNIT=3330,MODEL=11,                                *
              FEATURE=(SHARED,ALTCTRL),                              *
              ADDRESS=(4A0,16),UNITADD=A0,CUNUMBR=(42,43)
*IOCP
        CNTLUNIT  CUNUMBR=44,UNIT=3880,UNITADD=((E0,16)),            *
              PATH=(12),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=45,UNIT=3880,UNITADD=((E0,16)),            *
              PATH=(0C,28),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,FEATURE=(ALTCTRL,SHARED),                *
              ADDRESS=(4E0,8),UNITADD=E0,CUNUMBR=(44,45)
*
*
********************* UNIT ADDRESSES 500-5FF ************************
*
        CNTLUNIT  CUNUMBR=50,UNIT=3480,UNITADD=((60,16)),            *
              PATH=(0E,04),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3480,FEATURE=(SHARABLE),OFFLINE=YES,          *
              ADDRESS=(560,02),UNITADD=60,CUNUMBR=50
        IODEVICE  UNIT=3480,OFFLINE=NO,                              *
              ADDRESS=(562,04),UNITADD=62,CUNUMBR=50
        IODEVICE  UNIT=3480,FEATURE=(SHARABLE),OFFLINE=YES,          *
              ADDRESS=(566,10),UNITADD=66,CUNUMBR=50
*IOCP
        CNTLUNIT  CUNUMBR=52,UNIT=3803,UNITADD=((80,16)),            *
              PATH=(04,0E),SHARED=Y
        IODEVICE  UNIT=3420,MODEL=8,OFFLINE=YES,                     *
              FEATURE=(9-TRACK,ALTCTRL,OPT1600),                     *
              ADDRESS=(580,5),UNITADD=80,CUNUMBR=52
        IODEVICE  UNIT=3420,MODEL=5,OFFLINE=YES,                     *
              FEATURE=(9-TRACK,ALTCTRL),                             *
              ADDRESS=(585,1),UNITADD=85,CUNUMBR=52
        IODEVICE  UNIT=3420,MODEL=8,OFFLINE=YES,                     *
              FEATURE=(9-TRACK,ALTCTRL,OPT1600),                     *
              ADDRESS=(586,10),UNITADD=86,CUNUMBR=52
```

```
*
*
********************** UNIT ADDRESSES 600-6FF ************************
*
        CNTLUNIT  CUNUMBR=60,UNIT=3830,UNITADD=((00,16)),            *
              PATH=(06),SHARED=N
        CNTLUNIT  CUNUMBR=61,UNIT=3830,UNITADD=((00,16)),            *
              PATH=(13),SHARED=N
        IODEVICE  UNIT=3330,ADDRESS=(600,16),UNITADD=00,             *
              FEATURE=(SHARED,ALTCTRL),CUNUMBR=(60,61)
*IOCP
        CNTLUNIT  CUNUMBR=62,UNIT=3880,UNITADD=((60,16)),            *
              PATH=(0A,1E),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=63,UNIT=3880,UNITADD=((60,16)),            *
              PATH=(14),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,ADDRESS=(660,16),UNITADD=60,             *
              FEATURE=(SHARED,ALTCTRL),CUNUMBR=(62,63)
*IOCP
        CNTLUNIT  CUNUMBR=64,UNIT=3880,UNITADD=((80,16)),            *
              PATH=(0A,1E),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=65,UNIT=3880,UNITADD=((80,16)),            *
              PATH=(14),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,ADDRESS=(680,16),UNITADD=80,             *
              FEATURE=(SHARED,ALTCTRL),CUNUMBR=(64,65)
*
*
********************** UNIT ADDRESSES 700-7FF ************************
*
        CNTLUNIT  CUNUMBR=74,UNIT=3880,UNITADD=((C0,16)),            *
              PATH=(0D),PROTOCL=S,SHARED=N
        CNTLUNIT  CUNUMBR=75,UNIT=3880,UNITADD=((C0,16)),            *
              PATH=(17),PROTOCL=S,SHARED=N
        IODEVICE  UNIT=3380,ADDRESS=(7C0,16),UNITADD=C0,             *
              FEATURE=(SHARED,ALTCTRL),CUNUMBR=(74,75)
*IOCP
        CNTLUNIT  CUNUMBR=73,UNIT=2835,UNITADD=((F0,16)),            *
              PATH=(03,0D),SHARED=N
        IODEVICE  UNIT=2305,MODEL=2,                                 *
              ADDRESS=(7F0,1),UNITADD=F0,CUNUMBR=73
        IODEVICE  UNIT=2305,MODEL=2,                                 *
              ADDRESS=(7F8,1),UNITADD=F8,CUNUMBR=73
*
*
********************** UNIT ADDRESSES 800-8FF ************************
*
        CNTLUNIT  CUNUMBR=82,UNIT=3880,UNITADD=((70,16)),            *
              PATH=(21),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=83,UNIT=3880,UNITADD=((70,16)),            *
              PATH=(2B,09),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,FEATURE=(SHARED,ALTCTRL),                *
              ADDRESS=(870,8),UNITADD=70,CUNUMBR=(82,83)
*IOCP
        CNTLUNIT  CUNUMBR=85,UNIT=3880,UNITADD=((C0,16)),            *
              PATH=(21,26),SHARED=N,PROTOCL=S
        CNTLUNIT  CUNUMBR=86,UNIT=3880,UNITADD=((C0,16)),            *
              PATH=(2B),SHARED=N,PROTOCL=S
        IODEVICE  UNIT=3380,CUNUMBR=(85,86),                         *
              FEATURE=(SHARED,ALTCTRL),                              *
              ADDRESS=(8C0,8),UNITADD=C0,
```

```
*
*
********************** UNIT ADDRESSES 900-9FF ************************
*
         CNTLUNIT  CUNUMBR=90,UNIT=3880,UNITADD=((70,16)),             *
               PATH=(1D),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,FEATURE=SHARED,ADDRESS=(970,8),CUNUMBR=90
*IOCP
         CNTLUNIT  CUNUMBR=93,UNIT=3880,UNITADD=((D0,16)),             *
               PATH=(1D,27),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3350,FEATURE=SHARED,ADDRESS=(9D0,16),CUNUMBR=93
*
*
********************** UNIT ADDRESSES A00-AFF ************************
*
         CNTLUNIT  CUNUMBR=A2,UNIT=3880,UNITADD=((70,16)),             *
               PATH=(1E,28),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,FEATURE=SHARED,ADDRESS=(A70,8),CUNUMBR=A2
*
*
********************** UNIT ADDRESSES B00-BFF ************************
*
         CNTLUNIT  CUNUMBR=B2,UNIT=3800,UNITADD=((01,4)),              *
               PATH=(04),SHARED=N
         IODEVICE  UNIT=3800,MODEL=3,FEATURE=BURSTER,CUNUMBR=B2,       *
               ADDRESS=(B01,3),UNITADD=01,
*IOCP
         CNTLUNIT  CUNUMBR=B3,UNIT=3811,UNITADD=((07,1)),              *
               PATH=(08,09),SHARED=N
         IODEVICE  UNIT=3211,CUNUMBR=B3,ADDRESS=B07,UNITADD=07
*
*
********************** UNIT ADDRESSES C00-CFF ************************
*
         CNTLUNIT CUNUMBR=C1,UNIT=CTC,UNITADD=((C0,32)),PATH=(15),     *
               SHARED=N
         IODEVICE  UNIT=CTC,FEATURE=370,                               *
               ADDRESS=(CC0,32),UNITADD=C0,CUNUMBR=C1,TIMEOUT=N
*
*
********************** UNIT ADDRESSES D00-DFF ************************
*
         CNTLUNIT CUNUMBR=D0,UNIT=3880,UNITADD=((20,16)),              *
               PATH=(18),SHARED=N,PROTOCL=S
         CNTLUNIT CUNUMBR=D1,UNIT=3880,UNITADD=((20,16)),              *
               PATH=(22,2C),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,CUNUMBR=(D0,D1),ADDRESS=(D20,8),          *
               FEATURE=(SHARED,ALTCTRL),UNITADD=20
*IOCP
         CNTLUNIT CUNUMBR=D2,UNIT=3880,UNITADD=((70,16)),              *
               PATH=(18),SHARED=N,PROTOCL=S
         CNTLUNIT CUNUMBR=D3,UNIT=3880,UNITADD=((70,16)),              *
               PATH=(22,08),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,FEATURE=(SHARED,ALTCTRL),                 *
               ADDRESS=(D70,8),UNITADD=70,CUNUMBR=(D2,D3)
```

```
*
*
********************* UNIT ADDRESSES E00-EFF ************************
*
         CNTLUNIT CUNUMBR=E0,UNIT=3880,UNITADD=((20,16)),              *
               PATH=(19),SHARED=N,PROTOCL=S
         CNTLUNIT CUNUMBR=E1,UNIT=3880,UNITADD=((20,16)),              *
               PATH=(23,2E),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,ADDRESS=(E20,16),UNITADD=20,              *
               FEATURE=(SHARED,ALTCTRL),CUNUMBR=(E0,E1)
*IOCP
         CNTLUNIT CUNUMBR=E2,UNIT=3880,UNITADD=((70,16)),              *
               PATH=(19),SHARED=N,PROTOCL=S
         CNTLUNIT CUNUMBR=E3,UNIT=3880,UNITADD=((70,16)),              *
               PATH=(23,2D),SHARED=N,PROTOCL=S
         IODEVICE  UNIT=3380,ADDRESS=(E70,8),UNITADD=70,               *
               FEATURE=(SHARED,ALTCTRL),CUNUMBR=(E2,E3)
*
*
*
************************************************************************
*******                    UNITNAMES                           *******
************************************************************************
         SPACE 3
SYSRDR   UNITNAME NAME=SYSRDR,UNIT=(012)
SYSPR    UNITNAME NAME=SYSPR,UNIT=(004,006,007,00C,B01,B07)
TAPE     UNITNAME NAME=TAPE,UNIT=((580,16))
DASD2305 UNITNAME  NAME=DRUM,VIO=NO,                                   *
               UNIT=(7F0,7F8)
DASD3350 UNITNAME  NAME=DISKAA,VIO=NO,                                 *
               UNIT=((0C0,8),(160,4))
DASD3351 UNITNAME  NAME=DISKBB,VIO=NO,                                 *
               UNIT=((080,2),(100,2),(110,2))
DASD3380 UNITNAME  NAME=DASD,VIO=NO,                                   *
               UNIT=((220,16),(270,16),                                *
               (660,16),(680,8),(688,8),                               *
               (7C0,16),(9D0,16),(A70,08),                             *
               (D20,8),(D70,8),(E20,16))
SYSD3380 UNITNAME  NAME=SYSDA,VIO=YES,                                 *
               UNIT=((220,16),(270,16),                                *
               (420,16),(470,16),(4A0,8),                              *
               (660,16),(680,8),(688,8),                               *
               (7C0,16),(9D0,16),(A70,08))
SYSQ3380 UNITNAME  NAME=SYSSQ,VIO=YES,UNIT=((220,16),(270,16),         *
               (7C0,16),(9D0,16),(A70,08),                             *
               (D20,8),(D70,8),(E20,16),(E70,8))
*
*
************************************************************************
*******                    NIP CONSOLES                        *******
************************************************************************
*

CONSOLES NIPCON DEVNUM=(3D1,361,3C4,3D4,3D8)
         NIPCON DEVNUM=(360,363,3D0,3C7,3C8,3CE)
```

Appendix E. Eligible Device Table Verification Service

The EDT verification routine, IEFEB400, verifies that an EDT constructed by MVSCP matches the devices defined by the UCBs in the system on which IEFEB400 is running. If a mismatch is found, an appropriate message is written to the hardcopy log and to the operator.

Notes:

- *Do not* attempt to verify an EDT that was built through the previously used SYSGEN process. Such an attempt will give erroneous results.
- IEFEB400 cannot verify individual teleprocessing units; only the device class is checked.

During the IPL process, MVS invokes the EDT verification routine (IEFEB400) to check the entries in the eligible device table currently in use against the current unit control blocks (UCBs).

You can also run IEFEB400 as a batch job by submitting JCL statements. IEFEB400 must always be run on a system that is using the I/O configuration for which the new EDT was built. For an example of the JCL you will need, see "Sample JCL for Executing IEFEB400" on page E-3.

EDT Verification Processing

The processing that occurs when IEFEB400 runs during IPL is slightly different than the processing that occurs when it runs as a batch job. The following topics identify the messages, return codes, and reason codes that result from the different conditions that IEFEB400 detects. For additional message information, see *System Messages Volume 2 IEC - ISG*.

Batch Processing Results

- IEFEB400 always issues message **IEF923I** to identify the EDT that it will verify.
- If EDT verification is successful, IEFEB400 issues message **IEF926I**.

If an error or mismatch occurs, IEFEB400 issues one of the following messages and return codes.

Figure E-1. IEFEB400 Batch Results

Message	Condition Detected	Action Taken	Return Code (See note)
IEF924I	The device type in a UCB does not match the corresponding EDT entry.	Verification processing continues	20
IEF925I	The EDT includes a device type for which there is no matching UCB with the correct device number.	Verification processing continues	20
IEF927I	The EDT has not been rebuilt to run on MVS/XA.	Verification processing stops	16
IEF929I	The EDT identifier specified on the EXEC statement PARM field is invalid.	Verification processing stops	24
IEF929I	The EXEC parameter was valid but the specified EDT could not be loaded.	Verification processing stops	28

Note: IEFEB400 stores the return code into register 15.

IPL Processing Results

- IEFEB400 always issues message **IEF923I** to identify the EDT that it will verify.
- If EDT verification is successful, IEFEB400 issues message **IEF926I**.

If an error or mismatch occurs, IEFEB400 issues one of the following messages or wait state codes.

Figure E-2. IEFEB400 IPL Results

Message	Condition Detected	Action Taken	Wait Code/ Reason Code
IEF924I	The device type in a UCB does not match the corresponding EDT entry.	Verification processing continues	none
IEF925I	The EDT includes a device type for which there is no matching UCB with the correct device number.	Verification processing continues	none
IEF927I	The EDT has not been rebuilt to run on MVS/XA.	System wait state	200/0001
IEF928I	There are devices defined in the EDT for which there are no corresponding UCBs in the I/O configuration being used.	System wait state	200/0002

Sample JCL for Executing IEFEB400

```
//EDTVER   JOB 'ACCTINFO','NAME',MSGLEVEL=(1,1).....
//STEP1    EXEC  PGM=IEFEB400,PARM=xx
//SYSPRINT DD    SYSOUT=A
//EDT      DD    DSN=SYS1.EDTLIB
//SYSABEND DD    SYSOUT=A
```

Notes:

1. In the example, the variable *xx* must be the identifier of the EDT to be verified.
2. The EDT to be verified must be a PDS/PDSE member. In the example, the DD statement named EDT identifies this PDS/PDSE.
3. The I/O configuration in use on the system where IEFEB400 is run must be the I/O configuration for which the EDT was built.

Figure E-3. Sample JCL for Executing IEFEB400

Index

E

F

G

H

I

J

M

N

O

P

R

S

MVS/ESA
MVS Configuration Program
Guide and Reference

READER'S
COMMENT
FORM

GC28-1817-1

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

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

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

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

cut along this line

Reader's Comment Form

Fold and Tape Please Do Not Staple Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department D58, Building 921-2
PO Box 950
Poughkeepsie, New York 12602-9935

Fold and Tape Please Do Not Staple Fold and Tape

Printed in U.S.A.

IBM®