z/OS

Hardware Configuration Definition Planning

Version 1 Release 12

GA22-7525-14

z/OS

Hardware Configuration Definition Planning

Version 1 Release 12

GA22-7525-14

Note

Before using this information and the product it supports, read the information in “Notices” on page 119.

This edition applies to version 1, release 12, modification 0 of IBM z/OS (product number 5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GA22-7525-13.

Contents

Figures

Tables

About this information

This information describes how to define an I/O configuration using the Hardware Configuration Definition (HCD) element of z/OS®. It includes information on planning an I/O configuration, and specifying the configuration at IPL or dynamically, for a running system.

Who should use this information

This information is intended for anyone who plans an installation's I/O configuration and creates and coordinates action plans to change an I/O configuration.

How to use this information

If you are using this information for the first time, you should read all of the sections. Later, you might refer only to the information you need for a specific task that you are performing.

This information contains the following sections:

- Chapter 1, "Introduction," on page 1 presents an overview of I/O configuration tasks using HCD and dynamic configuration.
- Chapter 2, "Planning your I/O configuration definition," on page 11 describes the decisions you must make when planning your I/O configuration.
- Chapter 3, "Specifying an I/O configuration at IPL," on page 73 explains the steps for defining your I/O configuration at IPL-time.
- Chapter 4, "Dynamically changing an I/O configuration," on page 83 explains the steps for changing your I/O configuration on a running system.

Where to find more information

For information about using the specific HCD panels, see *z/OS HCD User's Guide* and the HCD help. Also you might want to see the ITSO *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*, SG24-4037, for an example of defining an entire, real-life configuration. This primer is available separately from z/OS.

For information about using Hardware Configuration Manager (HCM) for defining your configuration, see *z/OS and z/VM HCM User's Guide*.

Where necessary, this information references information in other informations, using shortened versions of the information title. For complete titles and order numbers of the books for all products that are part of z/OS, see *z/OS Information Roadmap*. The following table lists titles and order numbers for informations available separately from z/OS.

Short title	Publication title	Order number
Enterprise System/9000 Operator Guide	*Enterprise System/9000 Models 190, 210, 260, 320, 440, and 480 Operating and Recovery Guide*	GA23-0375

Short title	Publication title	Order number
Enterprise System/9000 Operator Guide	*Enterprise System/9000 330, 340, 500, 580, 620, and 720 Operator Guide*	SC38-0085
Hardware Management Console Guide	*Hardware Management Console Operations Guide*	GC38-0470
IOCP User's Guide	*Input/Output Configuration Program User's Guide/ ESCON Channel-to-Channel Reference*	GC38-0401
IOCP User's Guide for IYP IOCP	*Input/Output Configuration Program User's Guide for IYP IOCP*	SB10-7029
Stand-Alone IOCP User's Guide	*Stand-Alone IOCP, Generation 5, Generation 6*	GC38-0458
Stand-Alone IOCP User's Guide	*Stand-Alone Input/Output Configuration Program User's Guide*	SB10-7032
ESCON CTC Reference	*ESCON Channel-to-Channel Reference*	SB10-7034
PR/SM Planning Guide	*PR/SM Planning Guide*	GA22-7236
Support Element Operations Guide	*Support Element Operations Guide*	GC38-3119
S/390 Parallel Enterprise Server - Generation 5 System Overview	*S/390 Parallel Enterprise Server - Generation 5 System Overview*	GA22-7158

Additional hardware informations are available for specific processors. For titles and order numbers of related informations, see the informationation that comes with your processor.

Information updates on the Web

For the latest information updates that have been provided in PTF cover letters and informationation APARs for z/OS, see the online information at:

```
http://www.s390.ibm.com:80/bookmgr-cgi/bookmgr.cmd/BOOKS/ZIDOCMST/CCONTENTS
```

This information is updated weekly and lists informationation changes before they are incorporated into z/OS publications.

The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required): http://publib.boulder.ibm.com/infocenter/zoslnctr/v1r7/index.jsp

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to mhvrcfs@us.ibm.com
2. Visit the Contact z/OS web page at http://www.ibm.com/systems/z/os/zos/webqs.html
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Building 707
   2455 South Road
   Poughkeepsie, NY 12601-5400
   U.S.A.
4. Fax the comments to us as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  z/OS V1R12.0 HCD Planning
  GA22-7525-14
- The topic and page number related to your comment
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative
- Call IBM technical support
- Visit the IBM zSeries support web page at http://www.ibm.com/systems/z/support/

Summary of changes

Summary of changes
for GA22-7525-14
z/OS Version 1 Release 12

The document contains information previously presented in *z/OS HCD Planning*, GA22-7525-13, which supports z/OS Version 1 Release 10.

New information

- With the I/O Autoconfiguration function, HCD allows users to discover undefined FICON® storage devices (DASD and tape) connected to the processor via a switch. For complete information, see *z/OS HCD User's Guide*.
- HCD supports the IBM® z10™ BC processor (processor type 2098-E10). This function is available as SPE since V1R10.
- For the z/TPF operating system, 3215 consoles are supported on channel paths of type OSC. IOCP needs to distinguish between OSC-3270 and OSC-3215 attachments. This is done through the CHPARM keyword on the CHPID statement. This function is available as SPE since V1R10.

Changed information

- Table 9 on page 96 is modified.
- Table 8 on page 94 is valid for spanned channels.
- The "Readers' Comments - We'd Like to Hear from You" section at the back of this publication has been replaced with a new section "How to send your comments to IBM" on page xiii. The hardcopy mail-in form has been replaced with a page that provides information appropriate for submitting readers comments to IBM.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes
for GA22-7525-13
z/OS Version 1 Release 10

The document contains information previously presented in *z/OS HCD Planning*, GA22-7525-12, which supports z/OS Version 1 Release 9.

New information

- **Support for special devices and special secondary devices:** HCD supports the definition of *special devices*— that is, devices defined in the alternate subchannel set (not subchannel set 0). By placing devices in the alternate subchannel set, you can keep some device numbers free for later use. For more information, see "Defining special devices and special secondary devices" on page 40.

- **Information for new functions:** For a description of the new functions in this release, select Option 9 *What's new in this release* from the HCD Primary Task Selection Panel. There you might also find information about any small product enhancements (SPEs) that might have been delivered after the publication of this document.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes for GA22-7525-12 z/OS Version 1 Release 9

The document contains information previously presented in *z/OS HCD Planning*, GA22-7525-11, which supports z/OS Version 1 Release 9.

New information

- **Support for new hardware:** HCD supports the IBM System z10 processor family.

  Selecting a 2097 processor type creates a maximum logical processor configuration. That is, HCD generates a processor configuration with four channel subsystems and 15 reserved partitions in each channel subsystem. Also, HCD generates two subchannel sets in each of the four channel subsystems and provides for the maximum number of devices in each subchannel set.

  HCD support for predefining a maximum configuration for the System z10 platform is available for z/OS 1.7 and later. For earlier releases, you must define the configuration manually before creating the production IODF.
- **Channel path type CIB for coupling over InfiniBand:** The use of InfiniBand technology is supported through the new channel path CIB to emulate coupling connections in the Host Communication Adapter (HCA). Support is provided on processor types 2094, 2096, and 2097 as point-to-point InfiniBand connections between CPCs. You can define this new channel path type as you would any other channel path type, using for example, the HCD dialog or I/O configuration statements. For more information, see "Defining InfiniBand connections in a sysplex" on page 24.

Deleted information

- Support for processor types 4381, 9021, and 9121 has been removed.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes for GA22-7525-11 z/OS Version 1 Release 9

The document contains information previously presented in *z/OS HCD Planning*, GA22-7525-10, which supports z/OS Version 1 Release 8.

New information

- HCD supports Server Time Protocol (STP), a time synchronization feature that is designed to help multiple System z® servers maintain time synchronization with each other. For more information, see “Defining Server Time Protocol (STP) links in a sysplex” on page 24.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Chapter 1. Introduction

This topic describes the process of defining and selecting an I/O configuration through the Hardware Configuration Definition (HCD) element of z/OS. It also introduces the concept of *dynamic I/O configuration*, which is a means of selecting or changing the I/O configuration on a running system.

What an I/O configuration is

An I/O configuration is the set of hardware resources that are available to the operating system, and the connections between these resources. Hardware resources typically include:

- Channels
- ESCON/FICON directors (switches)
- Control units
- Devices.

When you define an I/O configuration, you provide both physical and logical information about these resources. For example, in defining a device you provide physical information, such as its type and model, and logical information, such as the identifier you will assign to the device. Chapter 2, “Planning your I/O configuration definition,” on page 11 describes the information you need to provide for each resource.

You define an I/O configuration to both the operating system (software) and the channel subsystem (hardware).

With Hardware Configuration Definition (HCD), you perform the hardware and software I/O configuration processes through a single, interactive end-user interface. As you enter data, HCD performs validation checking to help avoid data entry errors before you attempt to use the I/O configuration.

The output of HCD is an I/O definition file (IODF), which contains I/O configuration data. An IODF can define multiple hardware and software configurations to the z/OS operating system. When you activate an IODF, HCD defines the I/O configuration to the channel subsystem or the operating system, or both.

Through the HCD activate function or the MVS™ ACTIVATE operator command, you can make changes to the current configuration without having to initial program load (IPL) the software or power-on reset (POR) the hardware. Making changes while the system is running is known as *dynamic configuration* or *dynamic reconfiguration*.

You select the I/O configuration when you:

- POR
- IPL
- Activate a dynamic configuration change.

IPL and activation require that you identify the IODF that contains the definition of your configuration. A data set called an *I/O configuration data set (IOCDS)* is used at POR. An IOCDS can be created from a configuration definition in an IODF. The

IOCDS contains the configuration for a specific processor, while the IODF contains configuration data for multiple processors.

I/O configuration definition process

When you dynamically change an I/O configuration, you can change the I/O definitions for both the hardware and software, or for the software only, as follows:

- **Both hardware and software:** Usually, you make simultaneous dynamic configuration changes to both hardware and software configuration definitions.
- **Software only:** In some cases, you might want to modify only the software control structures, such as the unit control blocks (UCBs) and the eligible device table (EDT).

In some cases, you might make dynamic configuration changes to only hardware (a logical partition), which can affect the hardware configuration for other partitions. For more information, see "Dynamically changing an I/O configuration in LPAR mode" on page 84.

Defining an I/O configuration to the software

Figure 1 introduces the process of defining an I/O configuration to the software. At IPL, the system reads an IODF and constructs UCBs, an EDT, and all device and I/O configuration-related control blocks. When you select and activate a new configuration, the system determines the changes needed to the UCBs and EDT based on comparing the current IODF and the newly selected IODF.

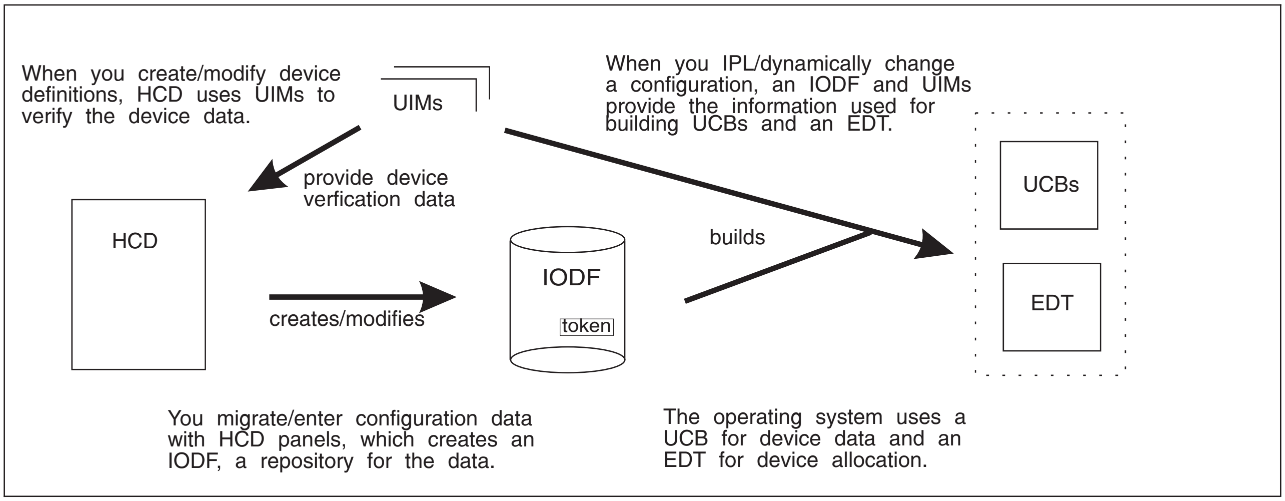

When you create/modify device definitions, HCD uses UIMs to verify the device data.
UIMs
When you IPL/dynamically change a configuration, an IODF and UIMs provide the information used for building UCBs and an EDT.
provide device verfication data
HCD
builds
UCBs
IODF
token
EDT
creates/modifies
You migrate/enter configuration data with HCD panels, which creates an IODF, a repository for the data.
The operating system uses a UCB for device data and an EDT for device allocation.

Figure 1. Defining an I/O configuration to the software

The major objects in Figure 1 are described in the sections that follow.

Hardware Configuration Definition (HCD)

Hardware Configuration Definition (HCD) allows you to define I/O configurations to both the software and hardware from a single, interactive interface. To define configurations to the hardware and software, you use HCD to create an input/output definition file (IODF).

You can use HCD to perform the following major tasks:

- Define new configuration data (an IODF or parts of an IODF)
- Activate configuration data
- View and modify existing configuration data

- Maintain IODFs (such as, copy, import, and export)
- Query and print configuration data
- Migrate configuration data
- Compare the active I/O configuration to a defined I/O configuration in an IODF. For a description of the software required for this verification, see *z/OS Planning for Installation*.

Input/output definition file (IODF)

An input/output definition file (IODF) contains information about the I/O configuration, such as:

- Operating system data
- Switch data
- Device data, including EDT definition
- Processor data
- Partition data
- Channel path data
- Control unit data
- Channel subsystem data.

Three types of IODF exist: a work IODF, a validated work IODF, and a production IODF.

Work IODF

A work IODF allows you to create a new I/O configuration definition or modify an existing I/O configuration definition. A work IODF provides a way to build or modify an IODF before you use it to activate a configuration. It is a working copy, not suitable for IPL selection or activated during dynamic activation. When a work IODF is ready to use, build a production IODF from it.

Validated work IODF

A validated work IODF satisfies all validation rules for building production IODFs. It may lack at least one physical channel identifier (PCHID). In cooperation with HCD and the CHPID Mapping Tool a validated IODF is required to accept new or updated PCHIDs. From such a validated work IODF, an IOCP input deck suitable for use with the CHPID Mapping Tool is generated. As soon as all PCHIDs are inserted or updated in the validated work IODF, the production IODF can be built. In HCD, you can use various methods to obtain a validated work IODF.

Production IODF

A production IODF defines one or more valid I/O configurations. A configuration in a production IODF can be activated dynamically or selected during IPL. Although you can build multiple production IODFs, only the one that is selected during IPL or activated during dynamic configuration is the active production IODF.

To change a configuration, create a work IODF from a production IODF. When you attempt to change a production IODF, HCD automatically copies the production IODF into a work IODF so you can make your changes. When changes are complete, build a production IODF from the work IODF.

You can organize your configuration data in an IODF in multiple ways. For example:

- You can define multiple operating system and processor configurations in one IODF. The configurations can share common device definitions; this reduces the maintenance effort when devices are changed.

- For a processor that supports logical partitions, you can define the configurations for the processor and the operating systems running in the logical partitions of that processor in one IODF.
- From the IODF representing the complete configuration (master IODF), you can create IODFs containing parts of the whole configuration (for example, a processor with all of its relevant operating systems). These IODFs can be distributed to a target system and used for activation. Note that the processor token does not change during the distribution process. You can also merge distributed IODFs in the master IODF. For information about handling large IODFs, see *z/OS HCD User's Guide*.
- You can define several IODFs for the same hardware, with different configurations for use in different circumstances. However, this approach is not recommended, because you might need to update multiple IODFs when you change the configuration.

Unit information module (UIM)

A unit information module (UIM) contains the information and rules that HCD uses to process I/O device definitions. When you create an IODF, HCD uses UIMs to validate the device definitions that you enter. The system invokes UIMs at IPL or during a dynamic configuration change to build UCBs.

UIMs are provided with IBM product software for devices that z/OS supports. For a non-supported device, you might be able to use a generic UIM or a UIM from a similar IBM device. Also, if you have an MVSCP UIM for a device, you can convert it to an HCD UIM.

For information about converting an MVSCP UIM to an HCD UIM, see *z/OS MVS Device Validation Support*.

Unit control block (UCB)

A unit control block (UCB) holds information about an I/O device, such as:
- State information for the device
- Features of the device.

You can access information in UCBs using UCB services, such as UCBSCAN and UCBLOOK. At IPL or dynamic configuration, UCBs are built from HCD device definition information in the IODF and UIMs. There is a UCB for each I/O device in a configuration.

Eligible device table (EDT)

An eligible device table (EDT) is an installation-defined and named representation of the I/O devices that are eligible for allocation. Using HCD, you define EDT information in an IODF. At IPL or dynamic configuration, information in the IODF and UIMs is used to build the EDT.

Defining an I/O configuration to the hardware

Figure 2 on page 5 introduces the process of defining an I/O configuration to the hardware. A configuration is defined to hardware when an IOCDS initializes the IOCDS information in the hardware system area (HSA) during a POR. A configuration is also defined to hardware when information in HSA is updated during a dynamic configuration change. More detailed descriptions follow the figure.

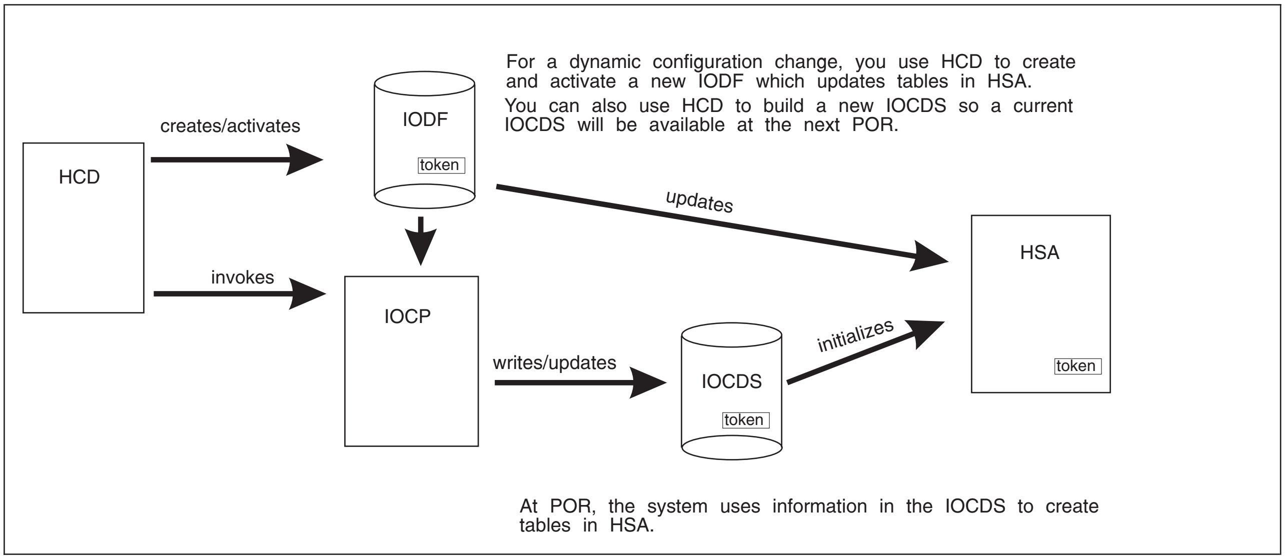

For a dynamic configuration change, you use HCD to create and activate a new IODF which updates tables in HSA.
You can also use HCD to build a new IOCDS so a current IOCDS will be available at the next POR.
HCD
creates/activates
invokes
IODF
token
updates
HSA
token
IOCP
writes/updates
IOCDS
token
initializes
At POR, the system uses information in the IOCDS to create tables in HSA.

Figure 2. Defining an I/O configuration to the hardware

I/O configuration program (IOCP)

I/O configuration program (IOCP) creates an IOCDS to define a configuration to the channel subsystem at POR. IOCP is shipped on the z/OS tape, and a stand-alone version is available on a processor.

You can specify at any time that you want an IOCDS written for a configuration defined in an IODF. It is a good practice to write an IOCDS when you activate an IODF so a current IOCDS is available for the next POR. Use the HCD Activate and Process Configuration Data panel and its option “Build IOCDS”. HCD invokes IOCP to write an IOCDS. IOCP builds a configuration definition from IODF information and stores the definition in an IOCDS so it is available when a POR is required.

Processors can have multiple I/O configuration data sets. On an HCD panel, you can specify the one that you want to update.

Notes:

1. You can use the option “Build and manage S/390® microprocessor IOCDSs and IPL attributes” to update the selected I/O configuration data set for one or more processors. For more information, see “Understanding a sysplex configuration” on page 21.
2. When you install a new processor that does not contain an IOCDS, you can use HCD to build an IOCP input deck, transfer the data set to tape, and run the processor's IOCP stand-alone program.

Input/output configuration data set (IOCDS)

An I/O configuration data set (IOCDS) contains information to define the I/O configuration to the processor complex's channel subsystem. The IOCDS is created by IOCP. A POR loads the IOCDS into HSA to initialize the hardware to reflect the IOCDS.

For most IBM processors, the IOCDS is stored on the support element hard disk. This includes the following processor families:

- IBM System z10

- IBM System z9®
- IBM zSeries® 990
- IBM zSeries 900
- IBM zSeries 890
- IBM zSeries 800.

For some earlier IBM processors, the IOCDS is stored on the processor controller DASD.

Hardware system area (HSA)

The hardware system area (HSA) contains tables that include information about the current configuration.

Ensuring that the software and hardware definitions match

Before your first dynamic configuration change, you must use HCD to create an IOCDS from the IODF that will be used for a subsequent IPL, then perform a POR using that IOCDS. The POR places information about the hardware configuration in the hardware system area (HSA). The same IODF must be used at IPL to define the software configuration.

Before you can perform a software and hardware dynamic change, your hardware and software definitions must match. When you use the same IODF to define your software and hardware configurations, the software and hardware definitions match.

However, there might be times when you want to change your software definition, for example, to temporarily run a test system. To change your software definition, you might perform a software-only dynamic change using a different IODF. In this case, your software and hardware definitions do not match and you cannot perform a full hardware and software dynamic configuration change.

To be able to make a dynamic hardware and software change, you can perform a software-only change using the IODF that was used to define the current hardware configuration. This IODF is the one used to define the software and hardware configuration at POR or at the last full software and hardware dynamic configuration change.

You could also perform a POR and the subsequent IPLs using the IODF with the current software configuration so the software and hardware definitions match.

Of course, the software-only dynamic change does not require an IPL. You just need to remember the IODF that you last used to define the software and hardware configuration so you can use it for the software-only change. This IODF is the one you used for the last POR or the last full software and hardware dynamic change. When you use this IODF for the software-only change, the software and hardware definitions match.

On the HCD View Active Configuration panel, you can view the IODF used for the last POR or software and hardware dynamic configuration change. To display the panel, choose the “Activate or process configuration data” option from the HCD primary menu and then the “View active configuration” option.

You can also use a token called the processor configuration token to find the IODF. (Do not confuse the processor configuration token with the MVS I/O configuration token. The MVS I/O configuration token enables programs that require configuration data to detect that dynamic changes have occurred.)

The processor configuration token exists in three places, as described in Table 1.

Table 1. Location of Processor Configuration Token

Where the Processor Configuration Token Exists	When the Token Is Placed There	How You Can View the Token There
In a production IODF, there is a processor configuration token for each processor definition.	When you change a processor definition and build a new production IODF, a unique token is placed in the production IODF.	View the token in the Processor Token field on the HCD View Processor Definition panel.
In the IOCDS.	When you build an IOCDS, the unique token is placed in the IOCDS.	View the token from the Hardware Management Console or use HCD.
In the HSA.	When you POR your processor or activate a software and hardware dynamic change, the system uses the IOCDS information to create tables in the HSA, which then contains the unique token.	View the token when you enter the MVS command 'D IOS, CONFIG' or view the token on the HCD View Active Configuration panel.

To determine the IODF that you last used for the software and hardware definition, view the token in HSA. Then view the tokens for each processor definition in the production IODFs until you find the IODF with the token that matches the token in HSA.

After you determine the IODF from the last POR or software and hardware dynamic configuration, use that IODF for a software-only change to have the software and hardware configurations match.

Attention: If the only IODF matching the hardware configuration in HSA is accidentally deleted, no dynamic changes are possible until after the next POR. For this reason, always keep a backup copy of your current IODF.

Changing hardware and software I/O configuration definitions

Changes to an I/O configuration include adding, deleting, and modifying the definitions of I/O components:

- You add an I/O component to the configuration definition by including the component's description in an IODF. You use HCD panels to add the component to a work IODF and create a production IODF from the work IODF.
- You delete an I/O component from the configuration definition by removing the component's description from an IODF. You use HCD panels to delete the component from a work IODF and create a production IODF from the work IODF.
- You change an I/O component in the configuration by changing the value of a characteristic in the component description in the IODF. You use the HCD panels to change component characteristics in a work IODF and create a production IODF from the work IODF.

Notes:

1. Some changes to device characteristics cause the UCB to be deleted and added again when the dynamic configuration change occurs. Message IOS502I reports the deleted and added device.
2. Any dynamic change that causes a device UCB to be deleted and added again also causes the device MIH time interval to be reset to the default MIH setting for its device class. To reestablish the previous MIH interval, enter the MVS command SETIOS MIH when the dynamic change is completed. For information about the SETIOS MIH command, see *z/OS MVS System Commands*.

In most cases, you will make simultaneous dynamic configuration changes to both hardware and software configuration definitions. However, in some cases making a change to only the software definition is useful. See “Making both hardware and software dynamic changes” and “Making software-only dynamic changes.”

Making both hardware and software dynamic changes

To make hardware and software changes, follow these steps:

1. Plan for changing the device configuration. For example, for adding devices, check that the device code was loaded at the last IPL. For deleting devices, verify that the device is not needed by any application running in the processor and that the device is offline. For more information, see “Making dynamic changes to I/O devices” on page 88.
2. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. For more information, see *z/OS HCD User's Guide*.
3. Activate the new IODF using the HCD Activate New Configuration panel or the ACTIVATE command.

   On the HCD Activate New Configuration panel, choose the “Activate hardware and software configuration” option.

   If you are using the ACTIVATE command, do **not** specify the SOFT keyword for hardware and software changes.
4. Use the HCD Build IOCDS task to write a new I/O configuration data set (IOCDS), if one does not already exist. You can make the new IOCDS the default for the next POR by issuing the ACTIVATE command with the ACTIOCDS=xx keyword. If necessary update the LOADxx members for each system to identify the new IODF.
5. Install the new I/O components that you have dynamically added through the new IODF. As required, perform the following tasks:
   - Configure the channel paths
   - Initialize devices, for example, format DASD with the volume serial
   - Enter the MVS VARY DEVICE command to make the device available to the system, for example, 'VARY dddd,ONLINE'.

For more information about making dynamic changes, see Chapter 4, “Dynamically changing an I/O configuration,” on page 83.

Making software-only dynamic changes

Although, you will usually make hardware and software dynamic changes, there are times when a software-only change is useful:

- When the software I/O configuration does not match the hardware I/O configuration. For example, when the IODF that you used for IPL does not contain the hardware I/O configuration used to create the IOCDS.

  You can use the software-only change to synchronize the software definition to the hardware definition so that a subsequent hardware and software change can be activated. When a software-only change switches the IODF, the new IODF can contain a processor configuration token that matches the current processor configuration token in HSA.
- When multiple logical partitions are running in LPAR mode. You make software-only changes in all partitions except one; in the remaining partition, you make a software and hardware change. (See "Dynamically changing an I/O configuration in LPAR mode" on page 84.)
- When z/OS is a guest operating system running under z/VM®.

  You cannot use dynamic configuration to make hardware changes in a z/OS guest system running under z/VM. However, you can use dynamic configuration to make software-only changes in the z/OS guest system. Thus, you can add new devices to a z/OS guest system using a z/VM interface (such as ATTACH or DEFINE), then use dynamic configuration to make software-only changes in the z/OS guest system. However, you must run this configuration change with the NOVALIDATE parameter.
- When z/OS is running on a processor that does not support dynamic configuration changes to hardware I/O configuration definitions. You can still make a software-only change, for example, to update the EDT or change the definition of a device from not shared to shared.

To make software-only changes, follow these steps:

1. Plan for your changes to the device definitions in the configuration. For example, for adding devices, check that the device code was loaded at the last IPL. For deleting devices, verify that the device is not needed by any application running in the processor. For more information, see "Making dynamic changes to I/O devices" on page 88.
2. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. See *z/OS HCD User's Guide*.
3. Activate the new IODF using the HCD Activate New Configuration panel or the ACTIVATE command:
   - On the HCD Activate New Configuration panel, choose either of the following options:
     – Activate software configuration only and validate hardware changes
     – Activate software configuration only

     Choose the "Activate software configuration only and validate hardware changes" option to validate that I/O components being deleted are not in use.
   - On the ACTIVATE command, specify the SOFT keyword for software-only changes.

For more information, see Chapter 4, "Dynamically changing an I/O configuration," on page 83.

Using HCM for I/O definition

Hardware Configuration Manager (HCM), an optional feature of z/OS, works with HCD to provide a set of graphical and text configuration reports. HCM complements HCD by adding information about the physical properties of the processors and I/O configurations to the logical information that HCD provides. For example, the physical information about processors and I/O configurations includes details about the ESCON® and FICON infrastructure such as cables, distribution panels, and patch panels.

HCM works as a client server application with HCD and uses the HCD validation process. Working with HCD, HCM combines logical and physical definitions into a single process. The HCM graphical interface allows users to define the configurations and to locate and display configuration objects and their connections. It also allows users to display operational data, such as system status information, and to operate on switches using an interface to the I/O operations component of System Automation for OS/390® (formerly known as ESCON Manager). The HCM graphical interface is helpful when planning configuration changes and doing problem determination.

For more information about HCM, see *z/OS and z/VM HCM User's Guide*.

Chapter 2. Planning your I/O configuration definition

Before you can create an IODF, you must first gather the necessary information.

This topic includes the following information:
- "Components of an I/O configuration"
- "Making decisions about your configuration" on page 13
- "Understanding a sysplex configuration" on page 21
- "Defining shared channel paths" on page 29
- "Specifying an I/O device number" on page 42
- "Specifying I/O device parameters and features" on page 46
- "Defining allocation preferences" on page 53
- "Migrating to subchannel sets" on page 62.

Components of an I/O configuration

Figure 3 shows typical components of an I/O configuration:

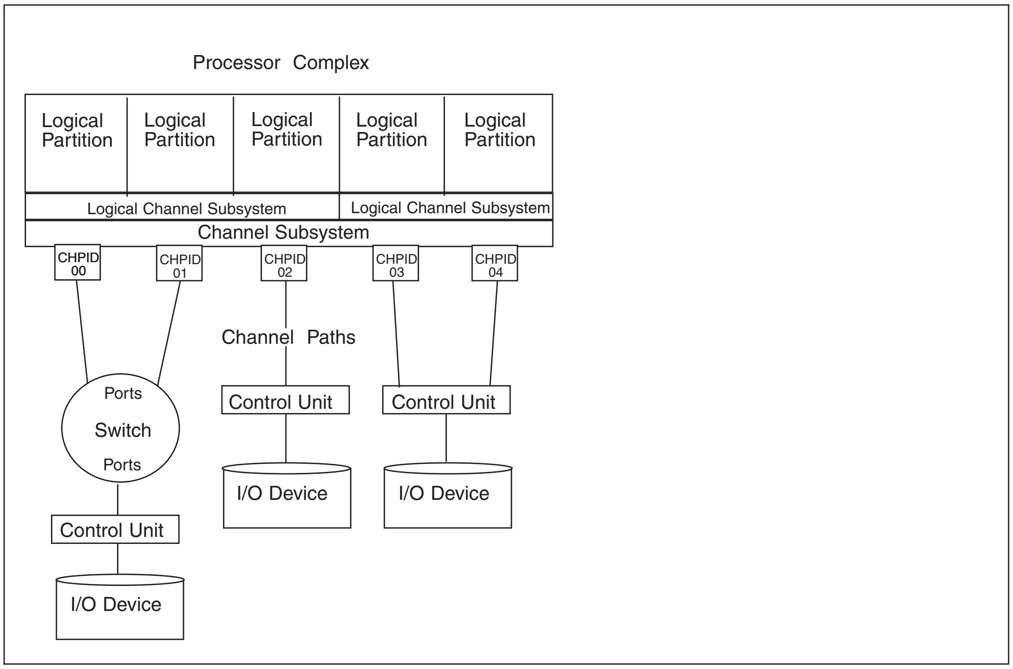


Figure 3. Configuration Example

When you define an I/O configuration, you specify:
- Operating systems
- Eligible device tables (EDTs) with esoteric device groups
- Processors
- Logical channel subsystems (for IBM zSeries 890 and later processors)
- Partitions (if processor is in LPAR mode)
- Channel paths on the processor(s)
- Switches (ESCON directors or Fibre Channel switches) attached to the channel path (if any)
- Control units attached to channel paths or switches

- I/O devices connected to the control units.

Logical partitions

The Processor Resource/System Manager (PR/SM™) feature allows a single processor to run multiple operating systems in logically partitioned (LPAR) mode. Each operating system has its own logical partition, which is a separate set of system resources including:

- A portion of storage (central, or central and expanded)
- One or more central processors. The processors can be dedicated or shared.

LPAR mode is established when the processor has had the PR/SM feature installed and has been divided into logical partitions.

Only LPAR mode (not basic mode) is supported on IBM zSeries 890, zSeries 990, System z9 or System z10 processors.

Logical control units for I/O devices

There are restrictions on a dynamic activation that adds or deletes a control unit based on how the hardware definition groups physical control units into logical control units. Each of the following is considered a logical control unit:

- A physical control unit with no devices attached
- A physical control unit that does not share any devices
- A set of physical control units that share any devices in common.

The type of processor determines the limit on the number of physical control units in a logical control unit. Figure 4 shows examples of logical control units.

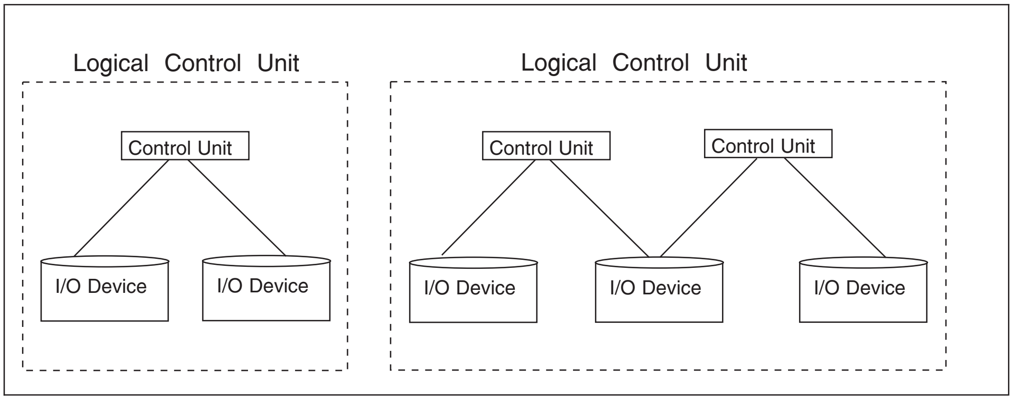


Figure 4. Logical Control Unit Examples

You cannot dynamically activate a change that causes one of the following:

- A physical control unit to be added to a logical control unit if the addition causes two logical control units (with devices attached) to merge into one logical control unit
- A physical control unit to be removed from a logical control unit if the deletion causes a split of one logical control unit (with devices attached) into two logical control units.

See *IOCP User's Guide* for requirements of defining control units to enable valid logical control units.

Channel path numbering

A channel path is logically identified by a two-digit hexadecimal number called a channel path identifier (CHPID). Some model processor complexes can have gaps in their channel path identifier (CHPID) numbering scheme due to the mix and number of installed parallel and ESCON channels. See the *IOCP User's Guide* for a description of the restrictions this might cause.

Selection of the path to access an I/O device

You can specify a preferred channel path to an I/O device. When you do not specify a preferred path, the channel subsystem uses a rotation order for the initiation of I/O requests to a device.

The rotation order is based on the:
- Control unit number you specify for a control unit that attaches a device
- Order channel paths are specified on a control unit

The channel subsystem tries the first channel path you specify for each control unit that attaches to the device, beginning with the lowest numbered control unit. The channel subsystem then tries the second channel path, if specified, for each control unit that attaches to the device, beginning with the lowest numbered control unit. The channel subsystem repeats this process until all specified channel paths for all control units attached to the device have been tried. See *IOCP User's Guide* for more information.

Using HCD, you can supply the information to establish preferred channel paths and the rotation order. To specify a preferred channel path, when you define a device, you specify the CHPID for the channel path in the Preferred CHPID field on the HCD "Device / Processor Definition" panel. When you assign control unit numbers and specify channel paths for control units, you can consider the rotation order to control the selection path order for devices.

Making decisions about your configuration

Before you actually define your configuration, decide on and collect the general information about the configuration components in a configuration plan. HCD provides reports about the processor support and UIMs installed on your system. These reports include information about device types, their parameters and features, and current connectivity rules. The Supported Hardware Report and the I/O Definition Reference are available as batch reports. After you define your configuration, you can use the reports that HCD generates to track updates to your plan.

After you create your configuration plan, use the HCD panels to specify the information about configuration components. HCD provides interactive help to describe the information for each field. The panels also enable you to request a list of options for some fields. For example, for the Processor Type and Model field, you can display a list of the available processor types to choose from with the Prompt function in the HCD panel.

The plan outline that follows shows the order that you would most likely follow to define your configuration:
1. Define operating system data
2. Define switch data
3. Define processor, logical channel subsystem, partition, channel path data, and subchannel sets

4. Define control unit data
5. Define I/O device data
6. Decide about JES3 initialization stream checker data (if you are running JES3)

IBM zSeries and System z platform considerations

IBM System z and zSeries environments support the following:

- More than 15 logical partitions. Only LPAR mode is supported; you cannot run in basic mode on these processors.
- Multiple logical channel subsystems, which allow you to logically partition your physical channel resources to accommodate large-scale enterprise workload connectivity and high bandwidth demands.
  - The z890 processor supports up to two logical channel subsystems.
  - The z990 supports up to four logical channel subsystems, depending on z/OS release and z990 LIC level. See "z890 and z990 processor support by release and LIC level."

  Each logical channel subsystem can have up to 256 CHPIDs and 15 logical partitions. Note that even a z990 at a level that supports 4 logical channel subsystems can still only have a total of 30 logical partitions for the processor.
- On an IBM System z9 with z/OS V1R7 or later installed, you can define up to two subchannel sets per channel subsystem. In subchannel set 0 (SS0) you can define up to 63.75K devices. In subchannel set 1 (SS1) you can define 64K alias devices. For additional information on defining subchannel sets, see "Defining multiple subchannel sets" on page 18.
- Channels can be shared, or spanned, across logical channel subsystems. Spanned channels allow you to minimize the number of physical CHPIDs, switch ports and cables required. Channels that would otherwise be under-utilized could have their utilization increased by spanning them across logical channel subsystems. See "Defining spanned channel paths" on page 35 for more information.
- You no longer specify the expansion percentages for HSA on the CPC image profile on the hardware management console (HMC). The expansion percentages are replaced by the maximum number of devices value when you define each of the logical channel subsystems. If you want to change this value, you must POR the machine. The default for the maximum number of devices value for each logical channel subsystem is 63K, so plan carefully to avoid wasting HSA space. On an IBM System z9 or later processor, you have two maximum number device parameters; one for SS0 with the default of 63.75K, and another for SS1 with the default of 0.

z890 and z990 processor support by release and LIC level

IBM z890 and z990 processor support varies by release and LIC level.

z890 and z990 compatibility support allows your z/OS system to run on or coexist with systems running on the z890 or z990 processor. Compatibility allows you to define multiple logical channel subsystems and up to 30 logical partitions, but z/OS can only run in and perform a dynamic ACTIVATE of hardware changes for a logical channel subsystem of 0. For details, see "Dynamic I/O configuration in a multiple LCSS environment" on page 99.

z890 and z990 exploitation support lets you run your z/OS system on a z890 or z990 processor in a logical partition in any logical channel subsystem, and allows dynamic ACTIVATE of hardware changes for any logical channel subsystem.

- On a z890 or z990 exploitation level or later, or a z990 with the May 2004 version of LIC **and** HCD APAR OA03689 installed, you can define up to four logical channel subsystems and to define additional channel types as spanned.

- On a **z/OS V1R6** system running on either a z890 or z990 processor, you can predefine one or more reserved logical partitions, which you can later add to the I/O configuration using dynamic I/O to assign a name and resources. You can also disconnect the partition from any assigned resources using dynamic I/O. The partition will revert back to being a reserved logical partition — the space used by the partition is not returned to the available pool of storage. See "Dynamically adding and removing logical partitions to the I/O configuration" on page 86 for more information.

  Note that on processors prior to the z890 or z990, you cannot dynamically add or delete a logical partition without building a new IOCDS and performing a POR.

Defining operating system data

To define the operating system configuration you need to decide on:

- An eight byte EBCDIC identifier for the configuration
- For eligible device tables (EDTs) for I/O devices:
  – Decide how many EDTs to define
  – Specify a two digit identifier for each EDT, for example, 20
  – Decide how many esoteric device groups to define in each EDT
  – Specify a name for each group

See "Defining allocation preferences" on page 53 for a description of EDTs and esoteric groups. When you define a specific device to your configuration, you specify the esoteric group in which to include the device as described in "Defining I/O device data" on page 20.

Defining switch data

Using HCD, you can define both ESCON switches (ESCON directors, such as 9032–5) and Fibre Channel (FC) switches. You can define an FC switch as a FICON FC switch (type 2032) or as a generic FC switch (type FCS). The FICON FC switch supports an internal switch control unit and device whereas the generic FC switch does not.

To define switches in your configuration, you need to know the switch type (for example, 9032–3 or 9032–5). You need to decide on:

- Switch ID, such as 01
- Switch control unit number and switch device number: You must define a switch control unit and device to be able to use HCD and the I/O operations component of System Automation for OS/390 (formerly known as ESCON Manager) to update the switch configuration and to validate the switch data. When you specify a control unit number and device number in the HCD Add Switch panel, a control unit and device are defined to be the same type as the switch. Note that the FCS switch does not support a switch control unit and device.
- Description: text that you use to describe the switch

You also use HCD to define the connections of the switch ports to channels or control units. You can use HCD to define a matrix of all possible connections between any two ports as well. The matrix information can be migrated into HCD from an ISPF table containing a switch configuration from System Automation for OS/390, from an active ESCON director (ESCD), a FICON director, or from a saved ESCD file. Use the "Migrate Switch Configuration Data" option on the HCD Migrate Configuration Data panel.

Similarly, you can use HCD to generate a matrix containing the defined logical connections between channel paths and control units.

Defining switch data is optional unless a system running the I/O operations component of System Automation for OS/390 must reconfigure the switch.

Defining processor, logical channel subsystem, partitions, channel path data, and subchannel sets

To define processors in your configuration, you need to know:

- Processor type and model.
- For a central processor complex (CPC) controlled through the Hardware Management Console, you require the SNA address of its support element. For more information, see "Understanding a sysplex configuration" on page 21.
- Serial number: If you specify a serial number, the system uses the number to verify that it is updating the correct processor during IOCDS download. This verification does not occur for a CPC that has a SNA address defined in HCD.
- Support level of the processor, for example if it supports ESCON channels. If only one support level is available for the processor type/model, you do not have to specify the support level in HCD.

To define processors in your configuration, you must:

- Choose an 8-byte alphanumeric processor identifier: a name that you assign to identify the processor in HCD, such as RN01PROC
- Select a processor mode: BASIC or LPAR

  If a processor is in LPAR mode, you must define partitions in your configuration.
- If the processor supports multiple logical channel subsystems, you must specify the number of logical channel subsystems (at least one).
- Supply a description: text that you use to describe the processor.

To define multiple logical channel subsystems in your configuration, you require an IBM zSeries 890 or later processor.

Also, you must determine:

- How many logical channel subsystems to define. The number you are allowed to define depends on your processor:
  - If you have either a z990 processor with the pre-May 2004 version of Licensed Internal Code (LIC) or a z890 processor, you can define up to two logical channel subsystems.
  - If you have a z990 with the May 2004 version of LIC and HCD APAR OA03689, you can define up to four logical channel subsystems.
- Which partitions are assigned to each logical channel subsystem.
- Which control unit information, such as channel paths and link addresses are defined for each logical channel subsystem for a processor.
- What channel path access, candidate lists, and preferred paths to define for a logical channel subsystem.
- Which channel paths are defined to each logical channel subsystem. Note that when you define a logical channel path, you must associate it with a logical CHPID on the HCD panels.
- How many devices are allowed . The value you choose should include the number of devices you define in your current I/O configuration plus room for growth in your configuration.

On an IBM zSeries 890 or later processor, you no longer specify the expansion percentages for HSA on the CPC image profile on the hardware management console (HMC). The expansion percentages are replaced by the maximum number of devices value when you define each of the logical channel subsystems. If you want to change this value, you must POR the machine. The default for the maximum number of devices value for each logical channel subsystem is 63K, so plan carefully to avoid wasting HSA space.

- Whether to predefine SS1 for each logical channel subsystem (on an IBM System z9 or later processor). You can use SS1 only for parallel access volume alias devices. For more information, see “Defining multiple subchannel sets” on page 18 and “Deleting parallel access volume alias devices dynamically” on page 45.
- How to map CHPIDs across multiple logical channel subsystems. CHPIDs are not unique across multiple logical channel subsystems, so to map a logical CHPID to a physical channel, you must specify a physical channel ID (PCHID).

To define partitions, you must choose:

- Partition name, such as MVS01 and RTR01
- Partition number/MIF image ID.

  On a processor prior to the z890 or z990, the partition number you specify on the "Add partition" panel in HCD when you define your partitions is also used as the Multiple Image Facility (MIF) image ID. MIF enables logical partitions to share channel paths and coupling facility sender channel paths between logical partitions within a processor complex.

  On an IBM zSeries 890 or later processor, the partition number and the MIF image ID are not the same. When you define your partitions on one of these processor using the "Add partition" panel in HCD, the value you specify is the MIF image ID, not the partition number. The system uses information in IOCDS to assign a partition number when you POR the machine. Because the value you specify on the HCD panel is a MIF image ID, the value must be between X'1' and X'F' and be unique within the logical channel subsystem. See “Processors communicating through shared ESCON or FICON channel paths” on page 32 for more information.
- Whether a partition will run an operating system, a coupling facility, or either, depending on whether coupling facility support is loaded. See “Understanding a sysplex configuration” on page 21 for more information.
- Description: text that you use to describe the partition
- On a z890 or z990 processor with a supporting level of microcode, you must decide whether to predefine one or more reserved logical partitions. Doing so reserves the needed space for the extra partitions in the IOCDS and HSA. Later, as needed, you can assign the partitions a name and assign resources to them using dynamic I/O. For more information, see “Dynamically adding and removing logical partitions to the I/O configuration” on page 86.

To define channel paths in your configuration, you first make your physical layout decisions such as your CHPID numbers for channels, number of channel paths, channel path type, and switch information if channel path(s) are connected to a switch. The switch information includes entry switch ID, ports, and dynamic switch ID. Then you need to decide on:

- Operation Mode: If a processor is in LPAR mode, you can specify the operation mode, which indicates the access that logical partitions will have to a channel path. The operation modes that you can specify for channel paths in HCD are:
  - **Dedicated** allows only one logical partition to access a channel path.

  - **Reconfigurable** allows only one logical partition at a time to access a channel path, but you can reconfigure that channel path from one logical partition to another. You reconfigure a channel path using the MVS CONFIG CHP(xx) command.
  - **Shared** allows more than one logical partition to access a channel path simultaneously. For information about shared channel paths, see “Defining shared channel paths” on page 29.

    You can only specify shared mode when the support level of the processor has Multiple Image Facility (MIF) capability. See “Defining shared channel paths” on page 29 for a description of the advantages and considerations for using shared channel paths.
  - **Spanned** allows partitions in more than one logical channel subsystem to share the same channel.

    Note that not all types of channel paths can be defined as spanned. See “Defining spanned channel paths” on page 35.
- Access and Candidate Lists: These lists specify which logical partitions can access a channel path. See “Defining logical partition access to a channel path” on page 30 for information.
- Dynamic channel path management: If you are using dynamic channel path management, a subfunction of IRD, you must define which channels and control units will participate. See “Defining dynamically managed channel paths” on page 33 for information.
- Intra-CPC communication: If you are using HiperSocket Accelerator, see “Defining IQD CHPIDs and device types” on page 41 for information.
- Inter-CPC communication: If you are using a timing link, see “Defining Server Time Protocol (STP) links in a sysplex” on page 24 for information.
- For a sysplex that uses the coupling facility, you need to decide on the connections between coupling facility receiver and coupling facility sender channels. The connections can be established using the 'View CF channel path connectivity' action on the HCD 'Channel Path List' panel. See “Understanding a sysplex configuration” on page 21 for more information.
- On an IBM zSeries 890 or later processor, you must specify a two-byte physical channel ID (PCHID) for each external channel to map the logical CHPID to its physical channel. You can specify them manually or use the zSeries CHPID mapping tool. You can get the CHPID mapping tool from Resource Link™ at:

  `http://www.ibm.com/servers/resourcelink`

  Note that internal channels, such as IQD do not require a PCHID.
- On an IBM zSeries 890 or later processor, you can define a channel path that is not physically installed on the processor. These are called over-defined channels. You might create over-defined channel paths if, for example, you wanted to prepare a configuration for future channel card upgrades. See “Creating over-defined channel paths” on page 36.
- Description: text that you use to describe the channel path

When you define a channel path, you specify the processor to which the channel will be attached. Define the processor before defining the channel path that is attached to it.

Defining multiple subchannel sets

For an IBM zSeries or System z processor, when you define a logical channel subsystem, you specify the maximum amount of HSA space you want to reserve for subchannels/devices. This space is called a subchannel set and can contain up to 63K subchannels on a z890 (or later) processor. On an IBM System z9 (or later) processor, you can define multiple subchannel sets. Subchannel set 0 (SS0) can be used for any type of device, but subchannel set 1 (SS1) can only be used for parallel access volume alias devices. Use multiple subchannel sets to move parallel access volume alias devices to SS1, then define more physical devices to SS0. See Figure 5.

Restriction: Any LPAR prior to V1R7 that is IPLed in the same logical channel subsystem that uses subchannel sets, will not have access to any SS1 alias devices. For more information on defining parallel access volume devices, see “I/O device numbers for a parallel access volume” on page 43.

To create an alternate subchannel set for parallel access volume alias devices for a logical channel subsystem, you must predefine it using HCD. You can then add and delete devices using dynamic I/O. Predefining an alternate subchannel set reserves the needed space for extra devices in the IOCDS. Then later, as needed, you can define devices and connect them to the I/O configuration using dynamic I/O. See “Dynamic I/O configuration in a multiple LCSS environment” on page 99 for more information.

Use one of the following processes to add subchannel support:
- “Scenario: All LPARs are z/OS V1R7 or later” on page 62
- “Scenario: Some LPARs are pre-z/OS V1R7” on page 62

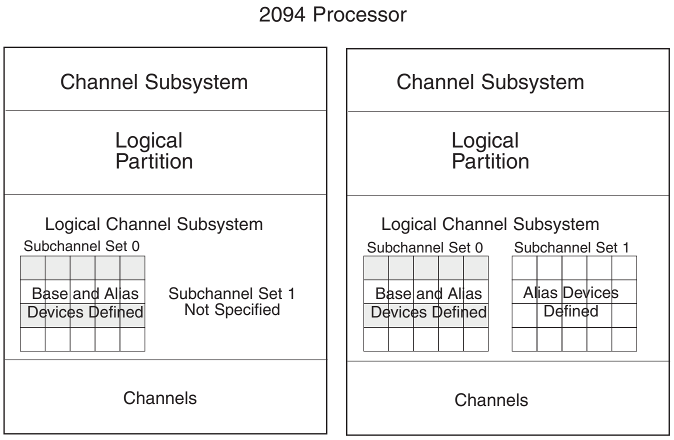


Figure 5. Example of multiple subchannel set configuration

You can use dynamic reconfiguration to move PAV alias devices to subchannel set 1. For an example, see “Example of migrating devices between subchannel sets” on page 63..

Defining control unit data

To define control units in your configuration, you need to know the control unit type and model, such as 9343-1. You need to decide on:
- Control unit number, such as 0060
- Connections to switches
- Description: text that you use to describe the control unit
- Processor(s) that the control unit will be connected to
- Logical channel subsystems that will have access to the control unit (for an IBM zSeries 890 or later processor)

- Information to attach the control unit to a channel path of the processor:
  - Channel path IDs
  - Link address
  - Unit address ranges that the control unit recognizes
  - Protocol
  - I/O concurrency level: classification of a control unit based on its ability to concurrently operate and control the activity of attached devices without causing loss of control or data.
  - Logical address, which is known as the CUADD value

Defining I/O device data

To define I/O devices in your configuration, you need to know the device type and model, such as 3420-7. You need to decide:

- The device number you want assigned to a device, for example, 0015. See "Specifying an I/O device number" on page 42 for considerations.
- Number(s) of the control unit(s) to which a device attaches.
- Device parameters and features for defining a device to an operating system, including whether the device supports dynamic configuration or whether a tape device is automatically switchable. See "Specifying I/O device parameters and features" on page 46 for more information.
- For esoteric device groups (that you named in the EDT as part of defining operating system data):
  - Which I/O devices you will include in each group
  - Which groups you want to make eligible for VIO. See "Defining devices as virtual I/O (VIO) eligible" on page 61.
- Any generic device types that you want to make eligible for VIO. See "Defining devices as virtual I/O (VIO) eligible" on page 61.
- Any changes you need to make to the device preference table for each EDT.
- The I/O devices that you will allow z/OS to use as NIP consoles.

  z/OS needs a console to initialize the system. The console can be a device that you identified as a nucleus initialization program (NIP) console through HCD. For each I/O configuration, you can specify through HCD a list of the devices that can be used as a NIP console. During system initialization, z/OS uses as the NIP console the first device in the list that is online and attached to the system.

  If your processor has the console integration feature, you can use the system console to initialize the system, and you do not need to specify NIP consoles through HCD. For more information about the consoles that the operator uses to initialize a system, see *z/OS MVS Planning: Operations*.
- Device parameters associated with the processor, such as unit address and preferred channel path ID. This information includes whether you want to control logical partition access to devices when logical partitions get access through a shared channel path. If you do want to limit logical partition access, you specify that you want an explicit device candidate list when asked by HCD. See "Defining logical partition access to a channel path" on page 30.

Deciding about JES3 initialization stream checker data

Consider whether you need to create data that the JES3 initialization stream checker will use.

During the JES3 initialization process, you must define to JES3 the devices that JES3 manages. It is essential that the JES3 and MVS I/O configuration definitions be consistent. You can use the JES3 initialization stream checker to identify inconsistencies in the JES3 and MVS I/O definitions. The JES3 initialization checker looks for logical errors in the JES3 initialization statements.

The JES3 initialization stream checker can test for I/O definition consistency only if you supply data about the MVS I/O configuration definition. HCD enables you to write to a data set the MVS configuration data that the checker needs. You use the data set as input when you run the checker. For more information about running the initialization stream checker, see *z/OS JES3 Initialization and Tuning Guide*.

To write the configuration data to the data set with HCD, you choose the “Create JES3 Initialization Stream Data” option on the HCD Activate and Process Configuration Data panel. On the HCD Create JES3 INISH Stream Checker Data panel, you specify the following information:

- Name of the data set that you created to contain the MVS configuration data to be input to the JES3 initialization stream checker
- Identifier of the operating system configuration and EDT for which you want to create the data set.

Understanding a sysplex configuration

HCD offers you a single point of control for systems in a sysplex that includes a coupling facility for data sharing. A coupling facility is connected to the processors through channels. With HCD you can:

- Define the hardware and software configurations in a sysplex
- Manage IOCDS and IPL attributes across a sysplex
- Dynamically activate hardware and software configuration changes across a sysplex.

This topic provides overview information about HCD definitions required with systems in a sysplex that includes a coupling facility. It also describes using HCD to manage IOCDSs in a sysplex and configuring your tape devices in a sysplex. For information about how to dynamically activate configurations in a sysplex, see “Dynamically changing an I/O configuration in a sysplex” on page 87.

Defining coupling facility components in a sysplex

A coupling facility is a special logical partition that runs the coupling facility control code (CFCC) and provides high-speed caching, list processing, and locking functions in a sysplex. HCD enables you to specify whether a logical partition can be a coupling facility, operating system, or either on certain processors. You connect the coupling facility logical partition to a processor through the coupling facility channels as shown in Figure 6 on page 22.

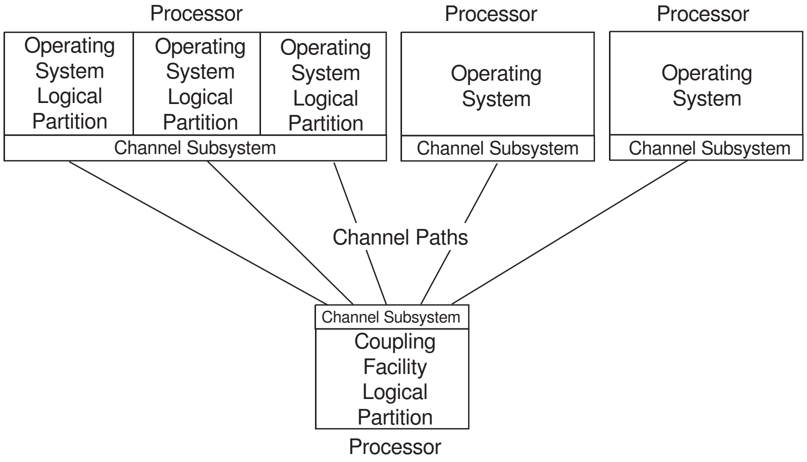


Figure 6. Example of a Sysplex

Using HCD, you define:

- A logical partition in which the coupling facility runs.
- Coupling facility (CF) sender channel paths and CF receiver channel paths or coupling facility (CF) peer channel paths. You define a coupling facility sender channel path to a logical partition running an operating system or a processor in basic mode running an operating system. You define a coupling facility receiver channel path to a logical partition running a coupling facility. On an IBM zSeries 890 or later processor, you define CF peer channels on both sides of a coupling facility connection. A CF peer channel contains both sender and receiver functions. However, it is not required that these functions are used on both sides.

  Note: With CF Duplexing, a CF logical partition can use the coupling facility sender function to communicate with another CF logical partition. As a result, you can define sender channel paths (CFS, CBS, ICS) besides the receiver channel paths (CFR, CBR, ICR) in a CF partition.
- Coupling facility control unit and devices. When you connect CF sender and CF receiver channel paths, or CF peer channel paths, HCD proposes coupling facility control unit and device numbers that must be defined for a CF sender channel. (CF receiver channels do not require control units and devices to be defined.) When you connect two CF peer channel paths, HCD proposes coupling facility control unit and device numbers on one or both sides of the connection depending on whether the sending function is used on only one or on both sides. You can override HCD's proposal to specify your own numbers for the control unit and device definitions.

  These coupling facility control units and devices have different characteristics than control units and devices. The ones defined as a result of coupling facility connections are referred to as coupling facility control units and coupling facility devices.

  Regardless of the CF channel path type used for the coupling facility connection, HCD uses:

  – Control unit type CFS and the device type CFS for the generated CF control units and CF devices specified for a CF sender channel path

- Control unit type CFP and device type CFP for the generated control units and CF devices specified for each of the CF peer channel paths

When you make connections between coupling facility channel paths, a coupling facility control unit and coupling facility devices are defined to represent the connections in the channel subsystem connection to the coupling facility. One coupling facility control unit is defined for each coupling facility that a processor connects to. For example, one control unit is defined when two logical partitions on a processor are connected to the same coupling facility partition. Two coupling facility devices are defined for each coupling facility path connection using CF sender and CF receiver channel paths. Seven coupling facility devices are defined for each connected CF peer channel path.

Coupling facility control units and devices are automatically defined by the CF sender-to-receiver or peer-to-peer connections that are established. You do not have to use the “Define Control Unit Data” option that you use to define control units and you do not have to use the “Define Device Data” option that you use to define I/O devices. See “Making changes to coupling facility devices and control units” on page 97 for information about changes to the coupling facility definitions.

The characteristics that distinguish coupling facility components are:
- UIMs are not needed for coupling facility devices
- Coupling facility devices are not represented by UCBs
- An EDT and dynamic allocation do not pertain to coupling facility devices
- You do not select a preferred channel path to a coupling facility device
- You do not vary coupling facility devices offline
- Channel measurement blocks (CMBs) do not apply to coupling facility devices
- You cannot specify a unit address for a coupling facility device
- The concept of logical versus physical control unit does not apply
- You cannot specify a logical address for a coupling facility control unit.

Coupling facility (CF) channels can be defined as Intersystem Coupling (ISC) channels or as Integrated Cluster Bus (ICB) channels. The channel path types used for ISC channels are CFS for the coupling facility sender channel, CFR for the coupling facility receiver channel, and CFP for the coupling facility peer channel. The channel path types used for ICB channels are CBS for the coupling facility sender channel, CBR for the coupling facility receiver channel, and CBP for the coupling facility peer channel.

For testing purposes, you can use a central processor complex (CPC) to simulate coupling facility channels.

You do not define coupling facility channel paths, a coupling facility control unit, or coupling facility devices for the logical partitions that use ICMF. ICMF automatically generates an appropriate configuration of integrated coupling facility paths, coupling facility control units, and coupling facility devices for the logical partitions to use.

With ICMF, if possible, avoid using the channel path identifiers, device numbers, or control unit numbers shown in Table 2 on page 24 for your I/O component definitions. ICMF uses these numbers when it simulates the coupling facility components.

Table 2. I/O Component Identifiers To Avoid When Using ICMF

Component Definition Information	Avoid Using These Numbers
channel path identifier	F0 and F1
control unit number	FFFF and FFFE
device number	FFFF, FFFE, FFFD, FFFC

As an alternative to ICMF, you can use Internal Coupling (IC) channels to allow an operating system image to communicate with a coupling facility image in the same CPC without requiring any hardware resources. Using IC channels is very similar to ICMF except for the following:

- CP microcode, not LPAR code, provides the coupling communication.
- Coupling channels can also be configured into either image along with Internal Coupling (IC) channels that need to be defined.

Except for channel packaging and providing connection information, IC channels follow the same configuration rules as real coupling facility channels (ISC or ICB).

Because you need no real hardware to perform IC functions, define IC channels with CHPID numbers that are not installed or are unusable because of other installed hardware.

Define the IC channels with the following types:

- CF sender channel with type ICS
- CF receiver channel with type ICR
- CF peer channel with type ICP.

Defining Server Time Protocol (STP) links in a sysplex

Server Time Protocol (STP) is designed to help multiple System z servers maintain time synchronization with each other, without the use of a Sysplex Timer®. STP uses a message-based protocol in which timekeeping information is passed over externally defined coupling links, such as InterSystem Channel-3 (ISC-3) links configured in peer mode, Integrated Cluster Bus-3 (ICB-3) links, and Integrated Cluster Bus-4 (ICB-4) links. These can be the same links that already are being used in a Parallel Sysplex® for coupling facility message communication.

An *STP link* is a coupling facility connection that serves as a timing-only link. With STP links, you can allow multiple servers to form a Coordinated Timing Network (CTN), which is a collection of servers and coupling facilities that are synchronized to a time value called Coordinated Server Time. Establishing an STP link between two processors does not require a CF partition; an STP link can be established between two OS partitions. For an STP link, HCD generates a control unit of type 'STP' on both sides of the connection. No devices are defined.

You can establish an STP link between two processors (for an IBM zSeries 890 or later processor). In the **Connect to CF Channel Path** dialog, select two CHPIDS capable for coupling facilities, and then specify the **Timing-only link** option to create an STP link.

Defining InfiniBand connections in a sysplex

HCD supports the use of InfiniBand technology to define coupling links and STP links between CPCs. This type of connection is supported on types 2094, 2096, and 2097 processors.

To define an InfiniBand connection, use channel path type 'CIB' to emulate a coupling connection in the Host Communication Adapter (HCA).

Reconfiguration of coupling facility receiver CHPIDs

You can reconfigure CF receiver CHPIDs to change the association between a z/OS image and a coupling facility (CF) logical partition without the overhead of a power-on-reset (POR). An operator command configures the CHPID to a different CF partition.

Reconfiguring CF receiver CHPIDs is intended for preplanned configuration changes that require the movement of CF channel capacity. This function is also useful in recovery situations; you can, for example, move channel capacity from lower priority work, such as test, to higher priority work, such as production.

There are restrictions on configuration definitions for reconfigurable CF receiver CHPIDs. The configuration rules for dedicated CF receiver CHPIDs, however, are not changed.

To understand the reconfiguration restrictions, you need to understand what "same z/OS image" means in the context of CF sender CHPIDs. The definition depends on whether the CF sender CHPID being connected is a dedicated or reconfigurable path, or a shared path:

- Both dedicated and reconfigurable CF sender CHPIDs belong to the same z/OS image if the CHPIDs have identical access lists. Any connects of these CF sender CHPIDs are made from the same z/OS image.
- Shared CF sender CHPIDs are treated as if they belong to the same z/OS image if there is at least one partition in common to all the CF sender channels. Any connects of these shared CF sender CHPIDs are made from the same z/OS image.

Restrictions: Reconfiguring CF receiver CHPIDs requires two CF partitions:

- Source CF partition that had initial access to the CF receiver CHPID at power-on-reset (POR)
- Target CF partition that has access after CF receiver CHPID reconfiguration

The CF partition specified in the access list is the source CF, and the CF partition specified in the candidate list is the target CF.

The following restrictions apply:

- When you define a reconfigurable CF receiver CHPID, specify only a single CF partition, the source CF, in the access list and only a single CF partition, the target CF, in the candidate list.
- You can define reconfigurable and dedicated CF receiver CHPIDs in the same CF partition:
  - You must define as dedicated any CF receiver CHPID initially accessed within a target CF logical partition.
  - You can define as dedicated or reconfigurable any CF receiver CHPID in a source CF logical partition as long as all connects from the same z/OS image are to CF receiver CHPIDs that are either all dedicated or all reconfigurable.
- You must reconfigure both CF sender CHPIDs and CF receiver CHPIDs together as part of a planned reconfiguration. If you reconfigure them separately, results are unpredictable.
- All connects from the same z/OS image to CF receivers with the same access list must have the same candidate list. All CF receiver CHPIDs that are connected

from the same z/OS image to the same source CF partition must be reconfigured to the same target CF partition.

- All connects from the same z/OS image to CF receivers with access lists that are different from any previous connection must have candidate lists that are different from any previous connection. All CF receiver CHPIDs that are connected from the same z/OS image to different source CF partitions must be reconfigured to different target CF partitions.
- All connects from different z/OS images to CF receiver CHPIDs with the same access list can have the same or different candidate lists. Not all CF receiver CHPIDs from the same source CF logical partition need to be reconfigured to the same target CF logical partition.
- All connects from different z/OS images to CF receiver CHPIDs with different access lists can have the same or different candidate lists. A target CF partition can have CF receiver CHPIDs reconfigured to it from many source CF partitions.

Reconfiguring coupling facility peer CHPIDs

A CF peer channel path may specify both z/OS images and CF logical partitions in its access and candidate lists. You can define a CF peer channel path as dedicated, reconfigurable, shared, or spanned. For a dedicated CF peer channel path, you can specify only one logical partition, running either an operating system image or a coupling facility. When you define a CF peer channel path as a reconfigurable, shared, or spanned, you can specify only one CF logical partition in its access list and one additional CF logical partition in its candidate list.

A coupling facility connection using CF peer channels must specify at least one z/OS image on one CF peer channel while the other CF peer channel must specify at least one CF logical partition. It is possible for each CF peer channel path of the connection to specify a CF logical partition.

With respect to the sender and receiver functions, the restrictions for reconfiguring CF peer channels in CF logical partitions are analogous to the CF receiver channel path restrictions described previously.

Managing IOCDSs and IPL attributes across a sysplex

HCD provides you with a single point of control for managing your I/O configuration data sets (IOCDSs). On the HCD panels, processors are referred to as S/390 microprocessor clusters. You create IOCDSs and update IPL parameters for each central processor complex (CPC) in a S/390 microprocessor cluster.

The Hardware Management Console consists of a hardware management console application (HWMCA) running on a PC and a LAN connected to the support element of each coupling facility and the support element of each CPC in an S/390 microprocessor cluster.

Figure 7 on page 27 shows an example of support elements connected to a Hardware Management Console.

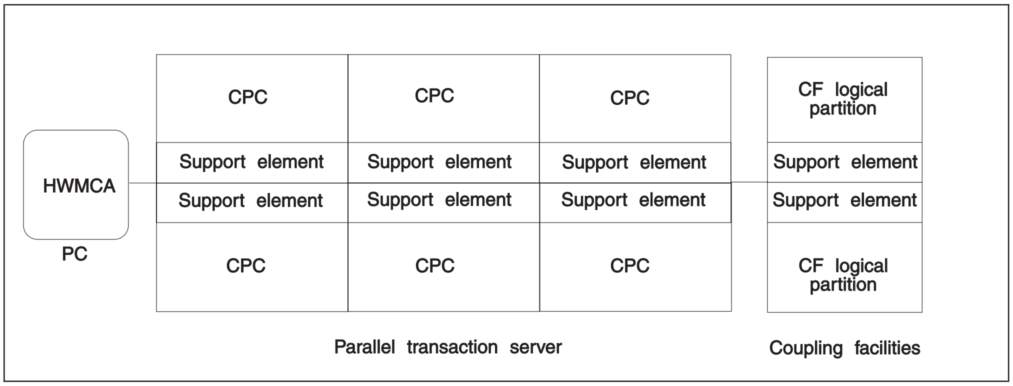


Figure 7. Support Elements connected to the Hardware Management Console

The system network architecture (SNA) address of the support element identifies the attached CPC to the Hardware Management Console.

HCD uses the SNA addresses to enable you to perform, from a single system, IOCDS management for any processors listed at the beginning of this topic. You can also use HCD to view and update the IPL address and LOAD parameter values for the next IPL. These values are used when IPLADDR(NEXT) and IPLPARM(NEXT) are specified on a load profile that is used to activate an operating system.

To enable these functions, you specify the SNA address of the support element on the HCD Add Processor panel when you define your processor. HCD provides a prompt facility that lists the SNA addresses for the currently active support elements. (For an S/390 9672 Parallel Transaction Server, you must define an IODF processor configuration for each CPC).

Notes:

1. Before you can use HCD to manage IOCDSs from a single system, you must use the Hardware Management Console to configure the processors. See *Hardware Management Console Guide* for information about using the Hardware Management Console.
2. For HCD IPL address and LOAD parameter changes to be effective, the processor activation profiles set up with the Hardware Management Console must allow dynamic changes. See *Support Element Operations Guide* for more information about CPC activation profiles.
3. To view IOCDS status in HCD, you need READ authority to the CBD.CPC.IOCDS resource in the RACF® FACILITY class. UPDATE authority is required to change IOCDS status information. To view IPL attributes, you need READ authority to the CBD.CPC.IPLPARM resource in the FACILITY class. UPDATE authority is required to change IPL attributes. When the CBD.CPC.IOCDS resource is defined in the FACILITY class, the operator message asking for approval to write an IOCDS is no longer issued when someone attempts to write an IOCDS. Only users with UPDATE authority to the CBD.CPC.IOCDS resource will be able to write an IOCDS.

   If a security product is not in operation, the HCD IOCDS management and IPL attribute functions are not available. For more information, see *z/OS HCD User's Guide*.
4. Only the processor that HCD is running in has to be operational to perform HCD IOCDS management and IPL attributes functions. As long as HWMCA

and the support elements are operational, HCD can perform these functions on the other support elements even if the processors attached to these support elements are powered off.

Using automatic tape switching

When you configure your tape devices in a sysplex, you must consider whether to have all your tape drives on one system, or whether you will spread them out. You can share tape devices across many systems, both with and without the need for manual intervention by the operator.

IBM offers the following ways for many systems to use the same tape devices:

- A **dedicated tape device**, which is varied online to one system at a time. For a second system to use that same device, an operator issues VARY commands (first VARY OFFLINE, then VARY ONLINE) to make the device available to the second system. In this information, tape devices are assumed to be dedicated unless specified otherwise.
- An **automatically switchable tape device**, which can be online to more than one system at a time. For one system to use an automatically switchable tape device, then another system to use the same device, an operator does not have to enter any VARY commands. When the system selects that device for allocation to a job step, the device is at once both assigned to the specific system and allocated to the job step. Automatically switchable tape devices require that the systems in the sysplex communicate with each other with the ATS STAR method available with z/OS V1R2 and later.

You can define a tape device as automatically switchable through HCD. For information about automatic tape switching, including benefits and set-up recommendations, see *z/OS MVS System Commands*.

Preparing for automatic tape switching

Preparing for automatic tape switching requires you to:

- Decide how many automatically switchable devices to define; see "Deciding how to use automatically switchable devices."
- Define tape devices as automatically switchable; see "Defining a tape device as automatically switchable" on page 48.

Deciding how to use automatically switchable devices: The system programmer determines the appropriate balance of dedicated devices and automatically switchable devices to meet the installation's needs. Factors that affect the decision are:

- The number of systems in the sysplex
- The number of devices that are eligible for automatic tape switching. Automatic tape switching requires "assign-capable" tape devices, such as 3490 or 3480 tape drives.
- The systems' workloads and the number of jobs requiring tape devices.
- Whether the installation has other ways to manage devices, such as JES3 or a vendor product.

Generally, tape devices that are already available to multiple systems and used periodically but not constantly are candidates for being defined as automatically switchable. These devices can be used more efficiently when available to any or all systems in the sysplex.

You might have good reasons to have a mixture of dedicated and automatically switchable devices in an installation. Consider the following example that shows a good way of splitting up your tape devices. In our example, a computing environment includes:

- A System z9® processor that runs the company's large batch applications
- An S/390 microprocessor cluster that runs a CICS® OLTP workload
- A coupling facility that connects the System z9 processor and the S/390 microprocessor cluster
- 60 3490 tape devices

Because of their heavy demand for storage, the batch jobs on the System z9 processor need the majority of the tape devices. The CICS applications have occasional and short-term need for tape devices. Rather than having an operator vary the tape devices online and offline as the systems' requirements for tape devices change, divide the tape devices so that the batch work and the CICS workload use the tape devices as follows:

- 50 of the devices are dedicated to the batch work on the System z9 processor.
- 10 of the devices are automatically switchable across the systems in the S/390 microprocessor cluster.

Operational considerations for automatic tape switching: The operations staff is responsible for varying the automatically switchable devices online. They also must respond to messages that describe problems caused during automatic tape switching.

For considerations, see *z/OS MVS System Commands*, which describes how to:

- Define automatically switchable devices
- Display information about automatically switchable devices
- Respond to automatic tape switching events

Defining shared channel paths

This topic provides background information that is helpful if you are defining shared channel paths in your I/O configuration.

Multiple Image Facility (MIF) capability enables logical partitions to share channel paths, such as ESCON, FICON, and coupling facility sender channel paths, between logical partitions within a processor complex. If a processor complex has MIF capability, and is running in LPAR mode, all logical partitions can access the same shared channel paths, thereby reducing the number of required physical connections. In contrast, if a processor complex does not have MIF capability, all logical partitions must use separate channel paths to share I/O devices.

Figure 8 on page 30 compares three unshared channel paths to a single shared channel path. In processor complex A, three logical partitions can access an I/O device through three unshared channel paths. In processor complex B, which has MIF capability, three logical partitions can access an I/O device through a single shared channel path.

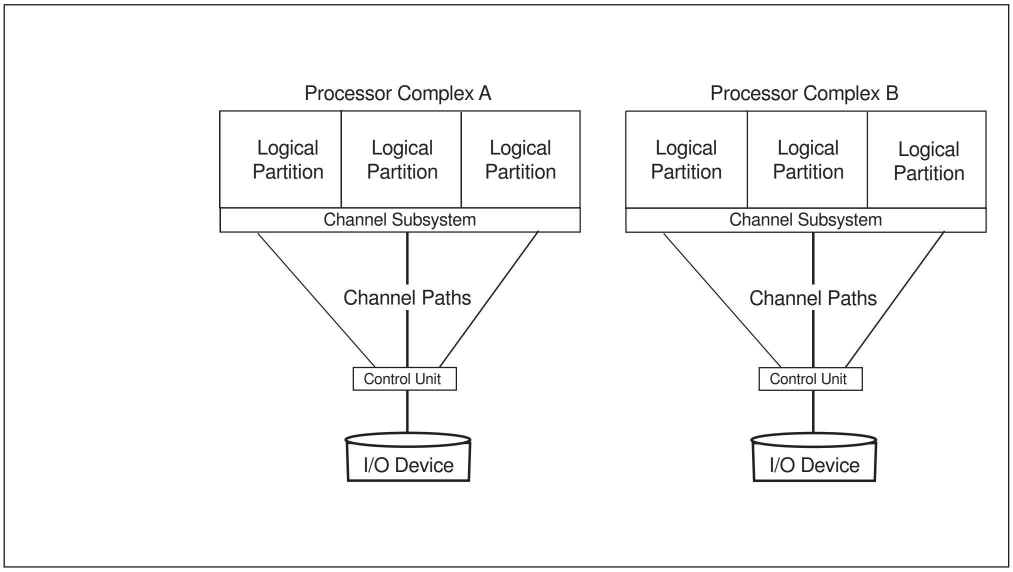


Figure 8. Shared Channel Paths

Considerations for sharing channel paths

Before using shared channel paths, observe the following considerations:

- ESCON and FICON channel paths can be shared; ESCON converter channel (CVC) paths and ESCON converter byte (CBY) channel paths cannot be shared. OpenSystem adapter (OSA) channels, OSA-Express channels (OSD and OSE), CF peer channels, coupling facility sender (CFS/CBS/ICS) channel paths, and ISD channel paths can be shared. CF sender channel paths can also be dedicated or reconfigurable. coupling facility receiver (CFR/CBR/ICR) channel paths can only be dedicated, though they can be defined as reconfigurable for the recovery scenario.
- Using shared ESCON and FICON channel paths does not reduce the number of logical paths needed at a control unit. A control unit requires a logical path for each logical partition that can access a shared channel path. For example, in Figure 8, the control units shown connected to processor complexes A and B both require three logical paths for the three logical partitions.
- The performance characteristics (for example, response time and throughput) as observed by each logical partition sharing a channel path depend on the total workload on the channel path. You can use multiple shared channel paths to provide adequate performance and recovery capabilities for critical I/O devices.

Defining logical partition access to a channel path

When a channel path is attached to a processor in LPAR mode, you can use HCD to specify which logical partitions have access to the channel path. You specify access by including a logical partition on the following channel path lists in HCD:

channel path access list

A logical partition that is on a channel path's access list can access the channel path when the logical partition is initially activated at POR. When a channel path is dedicated or reconfigurable, you specify one logical partition on the channel path access list. When a channel path is shared, you can specify more than one logical partition on the channel path access list.

channel path candidate list

A logical partition that is on a channel path's candidate list can eventually access the channel path. A logical partition on this list can access the channel path when the channel path is configured online to the logical partition

Note: HCD automatically considers a logical partition in an access list to be in the candidate list.

The operation mode of the channel (dedicated, reconfigurable, shared or spanned) determines the need for a candidate list. Dedicated channel paths are not reconfigurable so they do not need a candidate list. If no logical partitions are specified on an access list for a reconfigurable, shared or spanned channel path, then you specify logical partitions on the candidate list. Also for a shared or spanned channel path, if all the logical partitions are specified on the access list, they are all sharing the channel path initially so a candidate list is not necessary. However if not all the logical partitions are specified on the access list, a logical partition may be able to access the channel eventually if the logical partition is on the channel path candidate list.

When you define a device, HCD and MIF and IOCP also allow you to control logical partition access on a device level. A device might be attached to a control unit that uses a shared or FICON channel path. You can specify that only certain of the logical partitions sharing the channel path have access to the device.

To limit logical partition access, you specify that you want an explicit device candidate list when you define a device. On the device candidate list, you indicate the logical partitions that share the ESCON or FICON channel path(s) that can access the device. When you do not specify a device candidate list, all the logical partitions that share the ESCON or FICON channel path to the device can access the device. For a logical partition to use the device, the logical partition would have to be in the access list or candidate list of a channel path going to the device.

Sharing channel paths provides flexibility and reduced cost for an installation when defining an I/O configuration. An installation selects which I/O devices to access through shared channel paths, and can define and dynamically change which partitions have access to these channel paths and to the devices connected to them.

Notes:

1. When dynamically activating a configuration that adds a partition to the access list of a channel path, you must configure the channel path online to the partition, using the CONFIG command. You must use the CONFIG command at least once; afterward, PR/SM will configure the channel path to the partition during subsequent PORs. See *z/OS MVS System Commands* for information about the CONFIG command, and see *PR/SM Planning Guide* for information about the CHPID command and for a description of automatic configuration of channel paths to partitions.
2. Managed channels must be associated with a specific sysplex, and only images that are members of that sysplex can use them. The channel path access list and channel path candidate list cannot be specified for managed channels. For more information, see “Defining dynamically managed channel paths” on page 33.

Processors communicating through shared ESCON or FICON channel paths

An ESCON channel-to-channel (CTC) channel path has an ESCON CTC connection at one end, and either an ESCON (CNC) or FICON (FCV) channel connection at the other end. For any two processor complexes to communicate through an ESCON channel path, the ESCON CTC connection can be at either processor complex.

Processors at the IBM zSeries 900 level and later exploit the FICON capability to provide channel-to-channel host communication between processors. FICON CTC (FCTC) provides CTC communication at a higher bandwidth and with greater connectivity than with ESCON CTC. An FCTC connection to a processor earlier than IBM zSeries 900 (G5/G6 processors) is only possible if one of the two FC channel paths that perform the FCTC connection is defined on an IBM 2064 or later CPC.

ESCON or FICON CTC connections that are defined through a switch can be validated and reported in HCD (see the CTC Connection Report in *z/OS HCD User's Guide*. When using HCM, a separate dialog is provided for defining CTC connections. See *z/OS and z/VM HCM User's Guide*.

Using ESCON CTC connections

To communicate from a processor complex to a logical partition through an ESCON CTC channel path attached to a shared ESCON channel path, you must specify the logical address of the ESCON CTC (SCTC) control unit that represents the target system.

As shown in Figure 9 on page 33, the logical address is the number of the target logical partition. As explained in *z/OS HCD User's Guide*, you can specify partition numbers when defining partitions, and you can specify these partition numbers as the logical address for a CTC control unit.

For more information about communicating through ESCON CTC channel paths, see *ESCON and FICON CTC Reference*.

In the following example LP 1 has been assigned partition number 1, LP 2 has been assigned partition number 2, and so on.

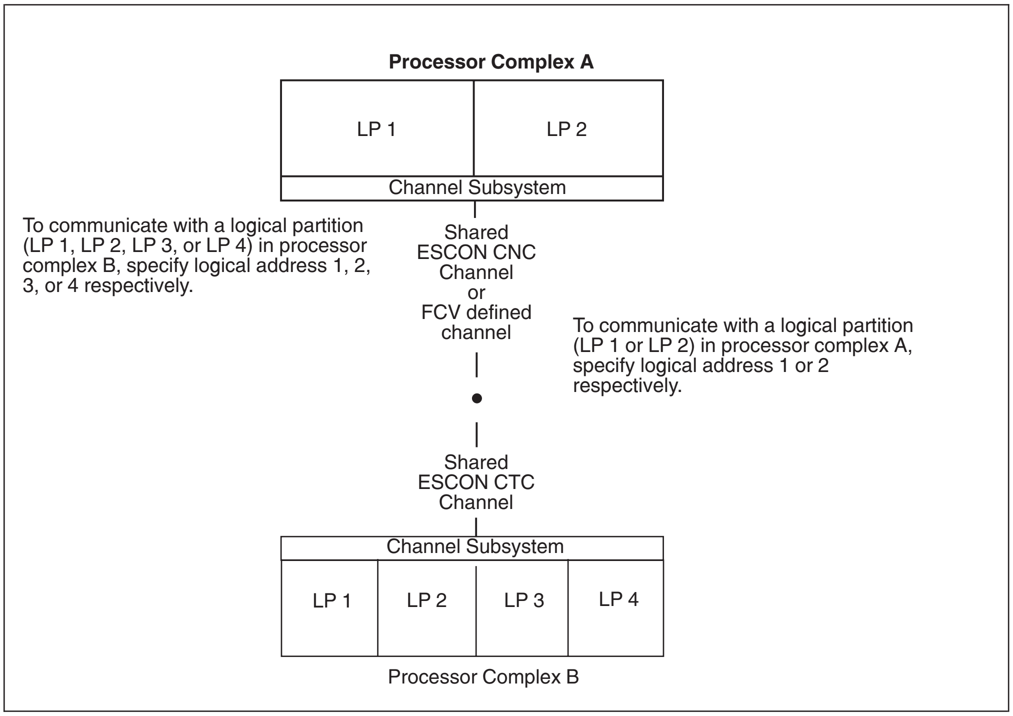

Processor Complex A
LP 1
LP 2
Channel Subsystem
To communicate with a logical partition (LP 1, LP 2, LP 3, or LP 4) in processor complex B, specify logical address 1, 2, 3, or 4 respectively.
Shared ESCON CNC Channel or FCV defined channel
To communicate with a logical partition (LP 1 or LP 2) in processor complex A, specify logical address 1 or 2 respectively.
Shared ESCON CTC Channel
Channel Subsystem
LP 1
LP 2
LP 3
LP 4
Processor Complex B

Figure 9. Logical Addresses and Logical Partition Numbers

You specify a logical address when you define a control unit in HCD. A logical address is not applicable for a coupling facility control unit.

Using FICON CTC connections

Unlike the ESCON CTC, the FICON CTC does not require that you explicitly define the CTC control unit function to one channel side or the other. The channel itself decides which side will contain the control unit function. That means, both sides of an FCTC connection use an FC channel path that is defined to one or more FCTC control units representing the target systems for the CTC connections. Like the ESCON CTC, a logical address is required when the FC channel path is attached to a shared FC channel path to identify the logical partition to be communicated with. When attached to an unshared FC channel path, the logical address must be zero or not specified.

FICON CTC communication within a single processor complex can be accomplished with a single FC channel path connecting to an FC switch, that is, both sides of the CTC connection are represented by the same single FC channel path. This results in:

- A reduction in the number of channels required
- A simplified configuration design
- A simplified configuration definition.

Defining dynamically managed channel paths

An integral part of Intelligent Resource Director (IRD), introduced in z/OS Release 1, is dynamic channel path management. Instead of defining a fixed number of channel paths to control units, this function lets Workload Management (WLM) move channel paths through the ESCON director from one control unit to another, in response to changes in the workload requirements. By defining a number of channel paths as "managed," they become eligible for this dynamic assignment.

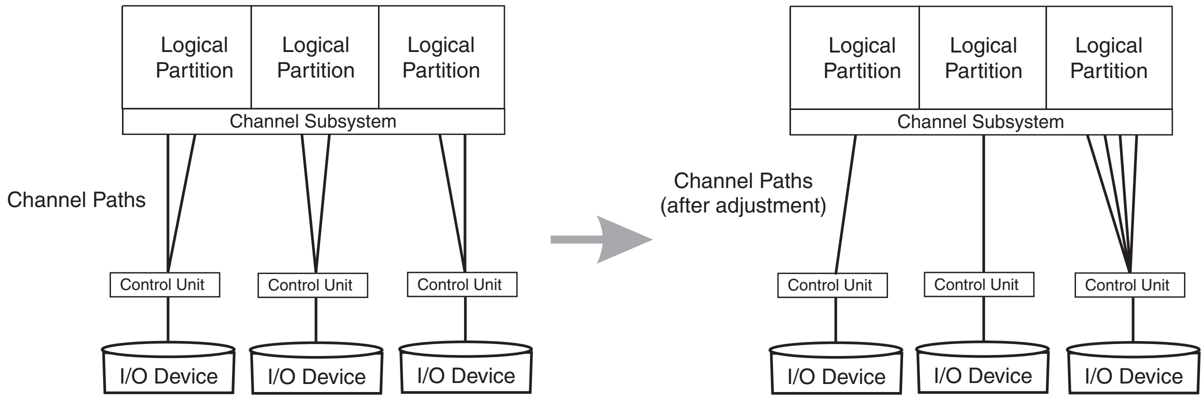


Figure 10. Dynamic Channel Path Management

Dynamic channel path management works in two modes: *goal mode* and *balance mode*. In goal mode, available only when WLM is running in goal mode, dynamic channel path management will manage the channel paths to best meet the performance goals of the most important work, even at the expense of less important work, if necessary. In balance mode, available whether WLM is running in goal mode or compatibility mode, dynamic channel path management will simply attempt to balance the responsiveness of all the DASD subsystems, moving dynamic channels as needed.

Dynamic channel path management is allowed only on an IBM 2064 (IBM zSeries 900) or later CPC. HCD ensures that no managed channel paths are connected to a CPC prior to this level.

IRD allows an installation to group logical partitions that are resident on the same physical server, and in the same sysplex, into an “LPAR cluster.” This gives Workload Management the ability to manage resources, both processor and DASD I/O, not just in one single image but across the entire cluster of logical partitions. To use dynamic channel path management, the CPC can be running in either basic or LPAR mode. In basic mode, a coupling facility is not needed, as there will be no logical partitions grouped together in an LPAR cluster. In LPAR mode, a coupling facility is required if you wish to use dynamic channel path management in any logical partition containing a system that is a member of a multisystem complex (even if the logical partition is the only member of that sysplex on this CPC). You do not need a coupling facility if all the logical partitions are running in XCFLOCAL or MONOPLEX mode.

Attention: Dynamic channel path management supports only non-synchronous control units, such as the Enterprise Storage Server®. A non-synchronous control unit transfers data with the host directly to and from the control unit cache. It does not support synchronous control units, nor control units that are synchronous on occasion, such as the 3990. The 3990 is a non-synchronous control unit most of the time, but there are situations in which it will transfer data to or from the device directly. Dynamic channel path management supports only 100% non-synchronous control units.

The IBM Redbook *z/OS Intelligent Resource Director* (SG24-5952) provides information about choosing the appropriate channels and control units for dynamic channel path management. The installations most likely to benefit are the following:

- Installations approaching the 256 channel limit per CPC (or 256 per logical channel subsystem on an IBM zSeries 890 or later processor)

- Smaller installations that do not have sufficient technical resources to design and monitor DASD subsystems
- Installations that experience large variations in channel utilization at different times of day
- Installations where the highest possible availability is an absolute requirement
- Installations with a large number of control units, and multiple control units, per channel
- Installations with a large number of EMIF channels.

After you have selected the appropriate channels and control units for dynamic channel path management, there are two HCD definitions that will need to be changed: the channel definitions and the control unit definitions.

To define a channel as being managed, you must specify YES in the “Managed” field on the “Add Channel Path” HCD panel. You must also specify a dynamic switch (in this case, the ESCON director) to which the channel is attached. It is recommended that you also specify the entry switch ID and entry port so that HCD can do consistency checking.

If you are running in LPAR mode, you must also define the name of the LPAR cluster to which a logical partition must belong in order to have this channel in its configuration. Specify this name in the “I/O Cluster” field. Finally, you must also define this channel as being shared. Note that unlike traditional shared channels which potentially can be shared by all logical partitions on a CPC, managed channels can only be shared by logical partitions in the same LPAR cluster.

To define a control unit as managed, you must make changes to its definition in the “Add Control Unit” HCD panel. Whereas non-managed channel paths (otherwise called “static” channel paths) are defined through the CHPID number, a managed path is defined by specifying an asterisk (“*”). The number of asterisks you specify will limit the number of managed channel paths per CPC (or managed channel paths per logical channel subsystem on an IBM zSeries 890 or later processor). The total number of non-managed and managed channel paths per CPC (or managed channel paths per logical channel subsystem) cannot exceed 8. On an IBM zSeries 890 or later processor, you must define the managed channel paths to the control unit in each logical channel subsystem.

Attention: You must define at least one non-managed channel path (which must be defined as shared) per control unit.

The control unit must be attached to a switch (in this case, an ESCON director) which in turn must be attached to managed channels.

See the *z/OS HCD User's Guide* for more information on using these panels to define managed channels and control units.

Defining spanned channel paths

This topic provides background information that is helpful if you are defining spanned channel paths in your I/O configuration.

A shared channel path can only be shared by partitions in a single logical channel subsystem. In contrast, a spanned channel path is one that can be used by partitions in more than one logical channel subsystem. You must use the same

CHPID value across all logical channel subsystems sharing the same spanned channel. However, logical channel subsystems that do not share a spanned channel can use that CHPID for other channels.

For more information, see “Dynamic I/O configuration in a multiple LCSS environment” on page 99.

Considerations for spanning channel paths

Before using spanned channel paths, you need to consider the following:

- Spanned CHPIDs still use access and candidate control lists to restrict access within the logical channel subsystem.
- Spanning reduces the number of channels that can be used. For example, on an IBM zSeries 890 or later processor, you can have up 512 channels available if you define two logical channel subsystems. However, each logical channel subsystem that has access to the spanned channel is using one of those 512 CHPIDs. So, if you have not defined any spanned channels, you have 512 channels available. If you define one channel as spanned across these 2 logical channel subsystems, you'll have only 511 channels available (512 - 2 used CHPIDs + 1 spanned channel). If you span all of your channels across all of your logical channel subsystems, you'll have 256 CHPIDs.
- Not all channel types can be defined as spanned. On a z990 processors with the pre-May 2004 version of LIC, only IQD and Internal Coupling peer channels can be defined as spanned. On a system at with HCD APAR OA03689 applied running on a z990 or z890 processor, you can span the following channels:
  - **Internal channels:** ICP and IQD
  - **Coupling facility peer and sender channels:** CBP, CFP, CBS, and CFS
  - **OSA channels:** OSC, OSD, and OSE
  - **Fibre channels:** FC, FCP
- Managed channels cannot be defined as spanned.

Creating over-defined channel paths

For an IBM zSeries 890 or later processor, you can define a channel path that is not physically installed on the machine. This is called an over-defined channel path. You might do this if:

- You want to migrate from a processor with more channels defined than the target z890 or later processor.
- You want to prepare your configuration for future channel card upgrades.

To distinguish an over-defined channel path from a physically installed one, use the character "*" for the PCHID value when creating it. An over-defined CHPID must adhere to all validation rules, but is not recognized when you use the HCD Build IOCDS function to create and download an IOCDS. In addition, they are not included:

- In the IOCP statements
- In the CONFIGxx parmlib member
- During a dynamic activation

If a control unit contains only over-defined channel paths, the entire control unit, including any attached devices, is omitted from the configuration to be activated.

When you later physically install an over-defined channel path, you must edit the CHPID and replace the "*" value with a valid PCHID. You must do this first even if you are going to use the CHPID mapping tool to assign the correct PCHID value.

FICON channels

FICON (fibre connection) channels increase the capacity of the channel subsystem; each FICON channel can replace up to eight ESCON channels.

Channel path definitions

You can define Fibre Channels in two ways:

- The FCV (Fibre Channel) channel that uses a FICON Bridge offers a migration path for ESCON CNC channels. Using the FICON Bridge Feature on the 9032-005 ESCON Director, you can attach ESCON devices to the FCV channel.

  An FCV channel path occupies eight port addresses on the switch. To model the FCV bridge within HCD, consider the following. Whenever you connect an eligible port address to an FCV channel path, you must set all other port addresses occupied by the FCV bridge to uninstall.
- The FC (Fibre Channel) channel requires a FICON interface at the control unit. FC channels can be attached directly to a control unit (point-to-point) or through a Fibre Channel (FC) switch.

Dynamic activation

There are no restrictions on dynamically activating an IODF that defines FCV or FC channel paths.

Migrating to FICON channels

Both HCD and HCM offer migration functions you can use to convert CNC channel paths to FCV channel paths, or to convert ESCON attachments (CNC or FCV) on the control unit to FICON attachments (FC).

Migrating from CNC to FCV channel paths

The HCM migration function includes three utilities. These utilities provide the ability to convert the CHPID type from CNC to FCV, to move ESCON port attachments effectively on the ESCON director, and to aggregate CNC CHPIDs onto an FCV CHPID.

The HCD migration function includes two dialogs, described in the following procedure for converting CNC channel paths to FCV channel paths. The procedure assumes that you have already decided which CNC CHPIDs you want to converge to a single FCV channel path and which ESCON director ports you will use for the FICON Bridge feature card.

The FICON Bridge feature occupies eight contiguous port addresses on the ESCON director, starting with port address x4 or xC. HCD requires you to connect the FCV channel path to the first port address and set the following seven port addresses to uninstalled.

After you have completed your planning, take the following steps:

1. Free the ports used by the FICON Bridge feature card.

   Disconnect the ports to be freed from the connecting units and connect them to a target port. At the same time, change the corresponding logical definition (such as link address or dynamic switch). Depending on your configuration, this step includes:

   - Moving CHPIDs that are connected to either port to another (free) port, and possibly to another switch, by updating the entry switch/entry port data in the channel path definition. When you move the channel path to a new switch, you must also change the dynamic switch, and move the connected control units to the new switch as well.

- Moving control units that are connected to either port to another (free) port, and possibly to another switch, by updating the connected switch/port data in the control unit definition. You must also change the link address. When you move the control unit to a new switch, you can move the connected channel paths to the new switch as well, and update the channel paths accordingly.
- Moving switch-switch connections to another free port. On the switch port list, disconnect the switch from the port and connect it to another port. To move a switch-switch connection, you might need to change the link address on control units that are accessed through a dynamic connection to the port to be freed, and move related channel path ports and control unit ports.

To perform this step in HCD, select the Move Ports action from the Switch List panel.

2. Change the switch matrixes to match the changed connections.

   If you are moving ports on the same switch, using the Port Move dialog, HCD automatically updates the switch configuration. If you are moving ports between different switches, you must update the switch configuration(s) of the target switch.

3. Define the FCV channel path.

   If the channel path you want to converge to the FCV channel path is already defined with a CNC CHPID, change the CHPID type from CNC to FCV.

   If a channel path is already defined with the ID selected for the FCV CHPID, and you cannot or should not aggregate this CHPID to the FCV channel path, change the ID of the existing channel path.

   Defining an FCV channel path might require you to change the IDs of other channel paths on the same channel card to conform to channel packaging rules.

4. Connect the FCV channel path to the FCV entry port.

   There is an entry port prompt on the Define Channel Path panel and on the Change Channel Path panel. This prompt shows the free ports on the entry switch, that are eligible to connect to FCV channel paths. For a switch configuration with an FCV channel connection, the following rules apply:

   - The FCV port must not have a dedicated connection defined.
   - If other FCV channel paths are connected to other ports, defining dynamic connections between any two of these ports is prohibited.

5. Aggregate other CNC channels to the FCV CHPID.

   To aggregate other CNC channel paths to the FCV channel path, you must (1) disconnect the control units connected to the CNC channel path from the CNC CHPID, then (2) connect the control units to the FCV CHPID.

   Depending on the access and candidate lists of the FCV channel path and the CNC channel path to be converged, you might have to update the device candidate lists. For example, the target FCV CHPID might be defined as shared, and the CNC channel path to be converged might be defined as dedicated or reconfigurable.

   If the devices connected to the channel paths that are aggregated have a preferred CHPID defined, you must update the preferred CHPID.

   To perform this step in HCD, proceed as follows:

   - On the Channel Path list, select the FCV channel path and apply the AGGREGATE CHPID action.

- On the Aggregate CHPID dialog, you can specify one CHPID at a time to be aggregated to the selected FCV channel path. Use the prompt to see the channel paths that are eligible for the aggregation. For conditions that must be met for a CHPID to be eligible for the aggregation, see the HCD online help.
- Set the ports following the FCV entry port, the ports that the FICON Bridge feature card will use, to uninstalled. If necessary, disconnect the aggregated CNC channel paths from the switch to free these ports.

Migration from CNC or FCV to FC channel paths on the control unit

You can seamlessly migrate from CNC or FCV to FC channel path attachments on the control unit (for example, on an ESS 2105 Control Unit) without varying the devices offline. You do this by the following steps:

1. Dynamically add an FC channel path to the existing list of channel paths that includes an ESCON channel path on the same (logical) control unit.
2. Configure the FC channel path online. The system routes the I/O requests via either the ESCON or the FICON channel attachments.
3. Configure the CNC/FCV channel paths offline so that I/Os are now directed via the FC channel path.
4. Dynamically delete the CNC/FCV channel paths.

With some restrictions, you can intermix ESCON and FICON channels on the same logical control unit. HCD supports the ESCON/FICON channel intermixture on the control unit according to the following rules:

- Intermixing FC channels with CNC and FCV channel paths should be temporary, and done only for migration purposes. This mixture is not recommended for production.
- Both the processor model and the control unit type must support the mixture of ESCON and FICON channel paths.
- If a control unit contains a mixture of CNC/FCV and FC channel paths, it is regarded as an ESCON control unit and validated according to the rules for ESCON control units, such as attachment limits.
- If a control unit attaches only to FC channel paths, it is regarded as a FICON control unit and validated according to the FICON control unit rules.

To determine the level to which your control units and devices support intermixing, see their device planning manuals. For those control units that do support intermixing, please consult their performance white papers for guidance on how to best configure channels from a performance perspective. To improve the balance of workload across channels in an intermixed environment, IBM recommends that you alternate ESCON (CNC and FCV) and FICON (FC) channel paths in the control unit using the following patterns:

- For 6 ESCON and 2 FICON, use a sequence of E, E, E, F, E, E, E, F
- For 4 ESCON and 4 FICON, use a sequence of E, F, E, F, E, F, E, F
- For 2 ESCON and 6 FICON, use a sequence of E, F, F, F, E, F, F, F

Open Systems Adapter Channels

Open Systems Adapter (OSA) channels allow:

- Direct connection of LANs using the Queued Direct I/O (QDIO) architecture
- Connection of LANs using non-QDIO architectures
- Attachment of TN3270E emulators for channels running the non-SNA DFT protocol (OSA-ICC)

- Communication between a host OS partition using Channel Data Link Control (CDLC) and the NCP emulation on Linux® using QDIO.

HCD supports the following OSA channel path types to use the OSA support facility:

- OSA QDIO channels (type OSD) are used for all LAN protocols that are supported by OSA features that use QDIO architecture.
- OSA non-QDIO channels (type OSE) are used for all LAN protocols that are supported by OSA features that use non-QDIO architecture.
- OSA ICC channels (type OSC) are use for TN3270E emulators. OSA-ICC is a replacement of the 2074 Console Support Controller and local, non-SNA 3174 control units.
- OSA NCP channels (type OSN) are used as replacement for ESCON channel communication between a host OS partition or VM guest (running VTAM® or TPF) and an SNA based 374x Network Control Program (NCP). The NCP function is emulated in Linux through the Communication Controller for Linux (CCL) on zSeries.

For more information about OSA channels, see *System z10, System z9 and zSeries OSA-Express Customer's Guide and Reference* or the zSeries IOCP User's Guide, SB10-7037.

Defining special devices and special secondary devices

You can use HCD to define special devices and special secondary devices. A *special device* is one that is defined in the alternate subchannel set (not subchannel set 0). A *special secondary device* is a Peer-to-Peer Remote Copy (PPRC) secondary device that is defined in the alternate subchannel set.

By placing devices in the alternate subchannel set, you can keep some device numbers free for later use. For example, you might choose to define a fixed block access (FBA) device, such as 3370, as a special device.

Unlike traditional devices, special devices and special secondary devices:

- Are not included in the system's eligible device table (EDT)
- Cannot be allocated through JCL or the DYNALLOC macro
- Cannot be varied online or offline by the operator.

Special devices and special secondary devices use the default MIH setting for their device classes. You cannot change the MIH interval for these devices through the MVS command SETIOS or through parmlib.

Special devices and special secondary devices are accessible only to authorized programs through the UCBSCAN and UCBLOOK macros. For more information, see *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*.

Defining a special device (3390S)

A device is considered special if it is defined in the alternate subchannel set. In HCD panels, a special device is shown as device type "3390S".

Observe the following considerations for a special device:

- The device is considered to be offline and in-use by the system.
- The IBM-provided DASD Error Recovery Procedure (ERP) is bypassed for an application that uses the device.

- The event notification facility (ENF) does not issue an event code to indicate the online or offline status of the device.
- To unbox a boxed special device, you can use the IOSODS service, that is: IOSODS OFF, then IOSODS ON.

Defining a special secondary device (3390D)

You can place PPRC secondary devices (or *special secondary devices*) in the alternate subchannel set and use the SWAP command to dynamically switch the devices concurrent to the applications using them. In HCD panels, a special secondary device is shown as device type "3390D".

For special secondary devices, observe the following considerations:

- MIH values are not exchanged during a swap.
- IPL and IODF devices cannot have secondaries in subchannel set 1.
- The devices are configured to come up offline.
- To unbox these devices, use the VARY PATH,ONLINE command.

For PPRC device pairs, observe the following considerations:

- The four-digit device numbers for primary and secondary devices must match.
- The associated primary device must be defined as a 3390B and have the same attributes as the secondary device.
- It is best to add or delete PPRC pairs at the same time, in a non-swapped configuration.

Defining IQD CHPIDs and device types

HiperSockets™ is defined as a channel connection with a channel path type IQD. The IQD (Internal Queued Direct Communication) channel type allows communication between logical partitions on a single CPC and provides high speed, low latency access to data. Support for IQD is available on IBM zSeries 900 and later processors with z/OS Communications Server.

Define the IQD CHPIDs to HCD along with the control units (type IQD) and devices (type IQD) that reside on them. Note that the control units and devices defined to either of these channel paths do not represent real I/O devices but data in a TCP/IP stack.

For an IQD channel path type, the maximum frame size to be used for IQDIO requests on that channel path must be specified. Valid maximum frame sizes supported are 16K (default), 24K, 40K, and 64K. See *z/OS Communications Server: IP Configuration Guide* for further information.

The following configuration rules apply for the IQD CHPID type:

- On an IBM zSeries 890 or later processor, up to 16 IQD CHPIDs can be defined per CPC and can be spanned across all logical channel subsystems and shared among all LPARs needing the function.
- On processors prior to the z890 or z990, you can define up to four IQD CHPIDs per CPC, to be shared among all LPARs needing the function. IQD CHPID numbers can range from X'00' to X'FF'.
- You can define up to 64 control units on each IQD CHPID. If more than one control unit is defined for an IQD channel path, a logical address is required for each control number. Control unit logical addresses can range from X'0' to X'F'.

- Up to 256 devices may be connected to an IQD control unit. The minimum value is 3. HCD has a default of 10 devices per control unit.Up to 256 devices can be connected to an IQD control unit. The minimum value is 3. HCD has a default of 10 devices per control unit.

  The maximum number of IQD devices that can be brought online at one time per z/OS image is 1023; however, because the z/OS Communications Server uses only a maximum of 10 online devices per IQD control unit for the HiperSockets function and you can define only 16 IQD CHPIDs per CPC, the number of usable devices per z/OS image is restricted to 160. Use caution when defining more than 1023 IQD devices. Because of the IOS device initialization algorithms, following an IPL, it is unpredictable which devices the system places online and which devices will be offline. When the 1023 device limit is reached, your attempts to issue VARY online commands for additional IQD devices are rejected with messages IOS577I and IOS579E. To bring additional IQD devices online, you must first VARY offline devices that were previously online. To identify which devices were not brought online, use the D M=CONFIG(xx) command.
- The total number of HSA subchannels for all IQD devices cannot exceed:
  - 3072 valid HSA subchannels on a processor prior to the IBM zSeries 890.
  - 12,288 valid HSA subchannels on an IBM zSeries 890 or later processor.

Restriction: You can define up to 256 IQD devices, but you cannot bring them all online simultaneously. The number of IQD devices you can bring online at one time is 160.

Specifying an I/O device number

This topic describes considerations for specifying a device number when you define an I/O device to your configuration.

A device number is the number you assign to identify a device in HCD. You assign a device number to each device to identify it in the configuration. A device number may be any hexadecimal number from X'0000' to X'FFFF'.

When you specify coupling facility channel path connections, HCD defines control units and devices to attach to the CF sender or peer channel. HCD assigns to each device a device number, which you can override. When assigning the device number, HCD searches to use the highest available number starting with FFFF and continuing in descending order. You might want to avoid assigning the higher range device numbers so they are available for HCD. In general, when operating a sysplex with a coupling facility, avoid higher range device numbers because of the way the system handles ICMF and CF sender or peer channels.

The last two digits of the device number can be the physical unit address of the I/O device. When you define a large number of I/O devices, situations might occur when it is difficult or undesirable to assign device numbers that contain the physical unit address of the device. When this occurs, you specify the physical unit address in the Unit Address field in the HCD Device / Processor Definition panel when you define a device.

You might want the same device number for two different devices across two partitions or even across processors. The devices have to be attached to different control units. To indicate whether devices identified by the same device number are the same or a different device, you can specify the serial number of the device to HCD.

For ease of tape device management, IBM recommends that you use the same device number on all systems to represent an automatically switchable device. If an automatically switchable device is a 3480 or a 3490 without the 'Read Configuration Data Capable' function, you **must** use the same device number.

Replicating I/O device numbers

When you define more than one device with the same device type, you can have HCD assign the device numbers for these devices. You specify to HCD the device type, the initial device number for the first device, and the number of devices of that type that you want defined. HCD defines the devices in the IODF using consecutive device numbers starting with the initial device number that you specified.

I/O device numbers for multiple exposure devices

The term *multiple exposure device* refers to an I/O device that can be *allocated* by a single device number but can be *accessed* by multiple device numbers from a single z/OS system.

I/O device numbers for a parallel access volume

Parallel Access Volume support provides the ability for a single operating system image to drive multiple concurrent I/O requests to a single device. A parallel access volume consists of a logical base device and associated logical alias devices. These logical devices are identified by device numbers and represented by unit control blocks (UCBs) in the system. When you request I/O to a parallel access volume, the request identifies the base UCB and the system uses the base UCB or one of its alias UCBs, depending on their availability, to initiate the request. You use HCD to define multiple device numbers for a control unit that provides parallel access volumes. Base device numbers and alias device numbers are defined independently.

A base device number is used to allocate the device, but other device numbers are needed to represent each alias UCB. The base device number is the one used when any of the following occurs:
- The device is allocated.
- The device is reconfigured with the MVS VARY command.

As shown in Figure 11 on page 44, when you define a parallel access volume in HCD, you define a control unit type that provides parallel access volume capability, for example '2105'. You define the devices for the control unit with parallel access volume device types. Base devices are defined using a base device type, for example '3390B' or '3380B'. Alias devices are defined using an alias device type, for example '3390A', or '3380A'. The device numbers are associated with unit addresses on the control unit using the 'unit address' parameter, which specifies the starting unit address for the set of devices being defined. The number of consecutive device numbers and unit addresses to be assigned is specified with the 'number of devices' parameter.

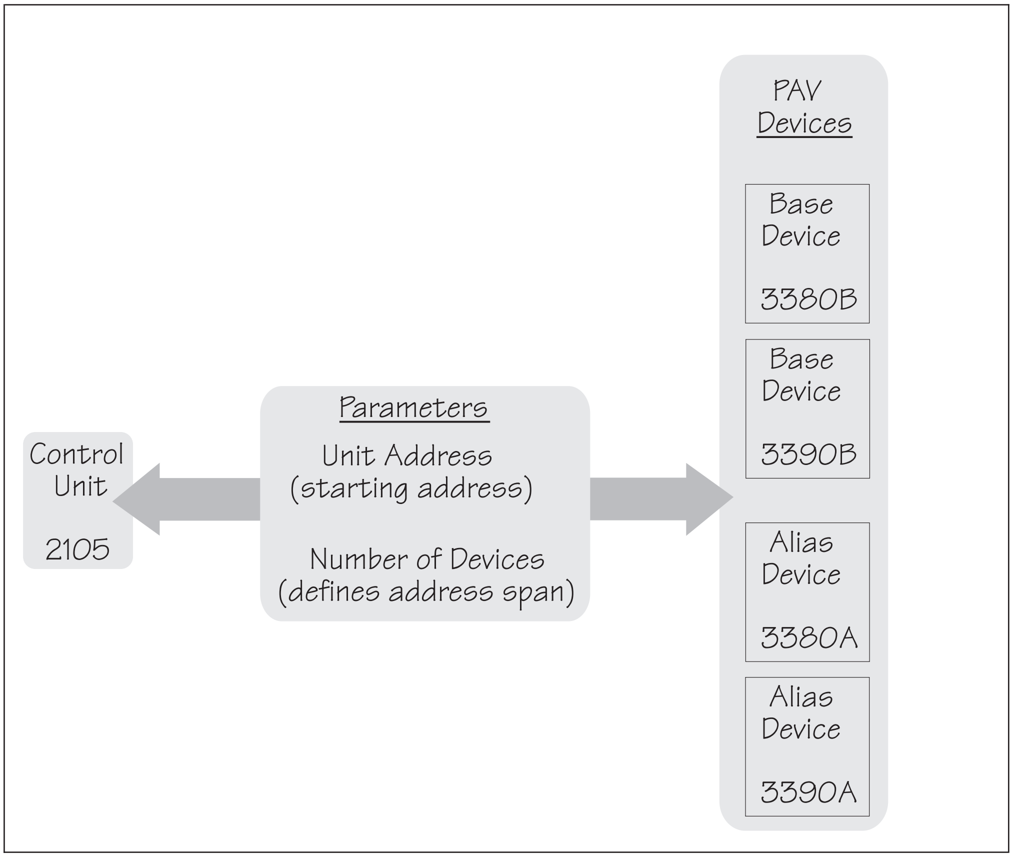


Figure 11. Control Unit - Parallel access volume device relationship

The number of aliases to be assigned to each base device is determined by the parallel access volume's configuration application (“ESS Specialist”) at IPL time or at the time a base device is varied online.

HCD provides the capability to specify device numbers for alias devices that are not contiguous with the device numbers for base devices on the same control unit. This capability can be useful in helping you to migrate a string of existing control unit and device definitions to parallel access volume control unit and device definitions. In this situation, you convert the existing device numbers to the base device numbers of the parallel access volume devices. The alias device numbers can then be specified to start with a device number outside the ranges used for any existing devices. For example, you can have one control unit with 64 devices numbered 1A0 through 1DF, and a second control unit with 32 devices numbered 1E0 through 1FF. You can change the first control unit to a parallel access volume control unit type that has 64 base devices and 192 alias devices, without disrupting the second control unit.

Table 3 shows the device numbers of the first parallel access volume control unit if you changed the starting alias device number to 21E0. Note that the device numbers for the second control unit are not shown because they are not disturbed.

Table 3. Example of Installation Assignment of Alias Device Numbers

Base 1A0(00)	Alias 21E0(40)	Alias 2220(80)	Alias 2260(C0)
Base 1A1(01)	Alias 21E1(41)	Alias 2221(81)	Alias 2261(C1)
Base 1A2(02)	Alias 212E(42)	Alias 2222(82)	Alias 2262(C2)
:	:	:	:

Table 3. Example of Installation Assignment of Alias Device Numbers (continued)

Base 1DF(3F)	Alias 221F(7F)	Alias 225F(BF)	Alias 219F(FF)
Values in parentheses are unit numbers on the control unit.			

Another benefit depicted in this example is that you can use four-digit numbers to identify aliases even when the base device numbers are three digits. Because most applications should not need to reference alias UCBs, applications that do not support four-digit device numbers are not affected. If your installation configuration definition lacks available three-digit device numbers and you define four-digit alias device numbers, you can migrate existing devices to parallel access volume devices without using any additional three-digit device numbers.

Deleting parallel access volume alias devices dynamically

You can dynamically delete parallel access volume alias devices to a base device using dynamic I/O in one of the following ways:

- If you are running on a pre-z/OS V1R6 system, you can only dynamically delete unbound parallel access volume alias devices. To unbind an alias, either vary it's associated base device offline or else delete it from the configuration application. For example, for a 2105 type control unit, the configuration application is the IBM TotalStorage® Enterprise Storage Server Specialist (ESS Specialist), which is a program with a Web-browser interface for configuring the ESS.
- If you are running on a system at the z/OS V1R6 level or later, you can dynamically delete both bound and unbound parallel access volume alias devices. This means that you can delete a bound alias device without first varying the base device offline or deleting it from the configuration application.

Workload manager and dynamic alias management

When you define parallel access volume device numbers, you can also choose to enable or disable dynamic alias management for that device. You do this by specifying **WLMPAV=YES** or **NO** in the HCD definition. YES is the default.

When dynamic alias management is enabled, workload manager (WLM) monitors the device performance and automatically reassigns alias addresses from one base to another to help work meet its goals and to minimize IOS queueing.

Note, however, that there is no consistency checking for dynamic alias management between different systems in a sysplex. If at least one system in the sysplex specifies WLMPAV=YES for a device, then dynamic alias tuning will be enabled on that device for *all* systems in the sysplex, even if other systems have specified WLMPAV=NO. It is recommended *not* to use dynamic alias management for a device unless all systems sharing that device have dynamic alias management enabled. Otherwise, WLM will be attempting to manage alias assignments without taking into account the activity from the non-participating systems.

Note, also, that you can specify WLMPAV=YES or NO on both base and alias devices. The WLMPAV setting on an alias device, however, is only meaningful when the alias device is bound to a base device that is offline, as follows:

- If the base device is *offline*, then only alias devices with WLMPAV set to YES will be reassigned to other base devices.

  The WLMPAV setting on the base device itself is irrelevant when the base device is offline, for either “giving” or “receiving” aliases. (Even if WLMPAV was set to YES on the base device, it cannot have new aliases assigned to it, as it is offline.)

- If the base device is *online*, then the WLMPAV settings on the aliases are ignored, as follows:
  - – If WLMPAV is set to YES on the base device, then the aliases can be reassigned regardless of their WLMPAV settings.
  - – If WLMPAV is set to NO on the base device, then no aliases can be reassigned, regardless of their WLMPAV settings.

In order for dynamic alias management to be most effective, you should try to spread out your aliases in the initial definition. If one base device has several alias devices while other base devices have none, it will take more time for WLM to reassign the aliases appropriately. Ideally, you should have at least two aliases assigned to each base at the outset.

Dynamic alias management can be globally enabled or disabled for the entire sysplex via the “Service Coefficients/Service Definition Options” panel of the WLM ISPF application. See the “Defining Service Coefficients and Options” topic in *z/OS MVS Planning: Workload Management* for more information.

Specifying I/O device parameters and features

This topic provides information about defining a device to the operating system. You specify parameters and features of a device to the operating system in HCD, such as:
- Whether a device is considered online at IPL
- Whether a direct access storage device (DASD) is shared by other systems
- Whether a device definition can be deleted dynamically
- Whether the UCB for a device is to reside above 16 megabytes
- Whether a device is automatically switchable

HCD displays only the parameters and features that are applicable to the device type.

Defining whether an I/O device will be dynamic

Each I/O device type is represented by a unit information module (UIM), which is included in the product that contains the device support code. The UIM for a device specifies whether or not the device type supports dynamic configuration. If your installation includes device types that support dynamic I/O configuration, you can change your I/O configuration without performing a power-on reset or re-IPLing the system.

Using HCD, you can specify whether a device with device support code that supports dynamic is considered dynamic or not in the operating system definition. Defining a device as dynamic allows you to dynamically add, modify, and delete the device definition while z/OS is running.

A device's support code and definition determine its inclusion in one of the following categories:
- Static
- Installation-static
- Dynamic

Table 4 on page 47 describes the categories.

Table 4. Description of Device Category Capability

Category	**Description**	**Capability**	**Notes®**
Static	Device with device support code that does not support dynamic.	• Device cannot be dynamically added, deleted or modified in software configuration definition. Therefore the device is not available for use until the next IPL of z/OS. • Device can be added, deleted, and modified in the hardware configuration definition, but its channel path information cannot be modified.	Device types that do not support dynamic configuration include: • An IBM device type that z/OS does not support. • A device whose device type does not support dynamic configuration. See "Device support for dynamic configuration" on page 48.
Installation Static	Device support code allows dynamic but is defined as DYNAMIC "No"	• Device can be dynamically added to the software definition but cannot be modified or deleted dynamically. • Device can be dynamically added, deleted and modified in the hardware configuration definition, but its channel path information cannot be modified.	If you must delete or modify installation-static definitions, you can redefine the installation-static device to dynamic. See "Redefining the dynamic parameter for an I/O device" on page 48.
Dynamic	Device support code allows dynamic and is defined as DYNAMIC "Yes"	• Device definition can be dynamically added, deleted and modified in the software and hardware definition.	A device definition for a device defined as dynamic can be deleted or modified only if ***both*** of the following are true: • The device is offline (implies that the device is not allocated). • The device is not pinned. (Pinning is a technique that prevents deleting the unit control block (UCB) and related data structures of a device. See "UCB services considerations" on page 113.

Notes:

1. When you define devices as dynamic make sure that your application programs are using UCB services that support dynamic. See "UCB services considerations" on page 113 for more information. You can define devices as installation-static initially until you can update your affected application programs to use UCB services that support dynamic.
2. Installations that IPL with I/O definition files (IODFs) that define dynamic devices should use the IPCS subcommands FINDUCB, LISTUCB, and IOSCHECK to validate and display dynamic UCBs. Any other subcommands might not produce valid results. FINDUCB, LISTUCB, and IOSCHECK are also the subcommands to use for validating and displaying UCBs for devices that have changed from installation-static to dynamic through an ACTIVATE command.
3. Any dynamic change that causes a device's UCB to be deleted and added again, causes the device's MIH time interval to be reset to the default MIH setting for its device class. To reestablish the previous MIH interval, enter the MVS command SETIOS MIH when the dynamic change is completed. For information about this command, see *z/OS MVS System Commands*.

Device support for dynamic configuration

To determine which devices on your system support dynamic I/O configuration, check the HCD device definition panel. If the panel indicates that the specific device can be defined as dynamic, the device supports dynamic configuration. For the most current information about device support for dynamic I/O configuration, you can also see the specific device publications.

You might need the following information about display devices and CTC connectors:

- **Display devices.** Display devices can be dynamically added or deleted, but cannot be used as multiple console support (MCS) consoles unless they are defined in the CONSOLxx member and have a UCB at the IPL.
- **Channel-to-channel (CTC) connectors.** CTCs can be dynamically added or deleted, but cannot be removed from global resource serialization rings, unless the CTC is managed by XCF. If so, your installation must stop XCF from using them before dynamically deleting the CTCs.

Redefining the dynamic parameter for an I/O device

An installation-static device that is part of an I/O configuration can be redefined from installation-static to dynamic. After you redefine the device from installation-static to dynamic, you can dynamically modify or delete the device from the configuration.

When you change a device from installation-static to dynamic, its unit control block (UCB) address changes. Programs on your system that save UCB addresses will encounter problems if they attempt to use the UCB that was valid when the device was installation-static. To minimize such problems, stop the program before making the dynamic configuration change. When you restart the program after the change, the program scans for and finds the new UCB representing the device.

Redefining a device from installation-static to dynamic does not delete the storage for the original UCB. You must assess the risk of how this may affect your applications and vendor products accessing the UCB.

Modifying or deleting an installation-static device requires two separate dynamic configuration changes:

1. Change the installation-static device to dynamic device (by activating a new IODF that defines the device as dynamic.) No other device characteristics for that device can be changed on the redefinition activation.
2. Delete or modify the dynamic device (by activating another IODF that contains the appropriate changes to the device).

Using the same steps, you can also redefine a device definition from dynamic to installation-static. You can do this if you mistakenly defined a device that was supposed to be installation-static as dynamic.

The UCB for an installation-static device is accessible through the limited UCB services. For more information, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

Defining a tape device as automatically switchable

A tape device is *dedicated* when it is capable of being online to only one system at a time. The VARY ONLINE command varies the device online to the issuing system and thereafter no other system can vary the device online until the first system varies it offline. In contrast to a dedicated device, an *automatically switchable* device can be online to one or more of the systems that participate in tape sharing within the same sysplex.

Besides HCD, a tape device can be defined as automatically switchable through:
- The VARY AUTOSWITCH command
- The IEEVARYD macro
- The ESCON manager.

Only through HCD, however, does the definition persist beyond the duration of the IPL.

For devices that support automatic tape switching, the AUTOSWITCH parameter appears on the HCD Define Device Parameters / Features panel. Specify a value of either YES or NO on the AUTOSWITCH device parameter.

You might want to change the AUTOSWITCH attribute of a device, for example, if your installation needs additional devices during peak workload hours. To change a device definition from dedicated to automatically-switchable, change the AUTOSWITCH parameter value from NO to YES.

When changed, the AUTOSWITCH parameter takes effect on the next IPL (the system does not check it during a dynamic software activation). To dynamically change the AUTOSWITCH parameter (that is, without an IPL), use the VARY AUTOSWITCH command.

For help in planning your use of tape devices, see "Using automatic tape switching" on page 28.

Defining the location of an I/O device's unit control block (UCB)

To conserve below 16 megabyte common storage, you can define a device's UCB above 16 megabytes. When a UCB is defined above 16 megabytes, programs requiring a 24-bit UCB address can still access the UCB. The system automatically captures the UCB at allocation. A captured UCB provides a 24-bit view into the above 16 megabyte UCB. A captured UCB resides in below 16 megabyte private storage for the duration of a device allocation. The captured UCB is released automatically at deallocation.

For information about accessing above 16 megabyte UCBs, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

Deciding whether to define above 16 megabyte UCBs

Because captured UCBs use private storage, you must carefully consider whether your installation should define above 16 megabyte UCBs. If you define above 16 megabyte UCBs, be aware that private storage is required for captured UCBs that are created in an address space when data sets are allocated. The system releases the storage when the data set is deallocated.

When you define above 16 megabyte UCBs, captured UCBs are required for certain applications, specifically those applications using data extension block (DEB)-based access methods. You can define above 16 megabyte UCBs for the applications that access them directly and for the applications that do not have private storage constraints when captured UCBs are created.

Capturing a UCB is page-oriented. Even if an address space captures only one UCB into private storage, the whole page of private storage is captured; a subsequent capture of a UCB that resides on the same page will not result in another page being captured to the address space. Other address spaces that need to capture the same UCB will have their own captured UCBs.

An exception to this would be if an address space requests a UCB to be captured to common (for example, through the macro CAPTCOM=YES) and then another address space requests a UCB on the same page to be captured to private. In this case, IOS returns to the second address space the common page address instead of the private page address. This makes sense because it is not necessary to keep multiple copies of the UCBs below 16 megabytes when they are already captured to common.

To determine the extent that you can take advantage of defining above 16 megabyte UCBs, understand whether the following situations apply to your installation. More than one case might apply depending on your installation's applications.

Case 1: The applications can access above 16 megabyte UCBs directly (for example, if you are using VSAM and dynamic allocation with the NOCAPTURE option) and the applications do not require 24-bit addresses for UCBs.

See *z/OS Planning for Installation* for information about subsystems that support above 16 megabyte UCBs.

Actions:
- Define devices with above 16 megabyte UCBs.
- Use dynamic allocation indicating UCBs are not to be captured.

Predicted Results:
- Constraint on common storage below 16 megabytes is relieved and private storage is not affected.

Case 2: The applications require 24-bit addresses for UCBs (for example, you are using DEB-based access methods) and your common storage is constrained. You can define enough above 16 megabyte UCBs to cause the private storage area to expand into the next megabyte (alleviating private storage constraint).

You can estimate whether you can define enough above 16 megabyte UCBs to cause the private storage area to expand into the next megabyte. At system initialization, the system rounds the amount of common storage to the next megabyte to create a megabyte boundary between below 16 megabyte common and private storage areas. You request common storage with the SQA and CSA parameters. To determine whether the private storage area will expand to the next megabyte, determine the amount of storage used for rounding and subtract it from one megabyte as shown in the following calculation:

```
1 MB - (actual CSA - requested CSA)
```

actual CSA - requested CSA

The amount of rounded storage that was allocated to create a megabyte boundary. Determine this amount by subtracting the amount of CSA requested with the CSA parameter in parmlib from the actual amount of CSA present in the system after IPL. You can determine the amount of actual CSA by viewing the RMF™ Monitor III common storage remaining table.

The result of this calculation is the amount of requested CSA as opposed to rounded CSA. If the amount of UCB storage you are moving above 16 megabytes is greater than this value, you have captured enough UCBs to cause the private storage area to expand to the next megabyte. If the amount is less than this value, see cases 3 and 4.

When determining the amount of UCB storage you are moving above 16 megabytes, remember that the size of a UCB is dependent on the device type. For example, a DASD UCB defined below 16 megabytes has approximately 128 bytes below 16 megabytes. The size of the prefix extension is not included because it already exists above 16 megabytes.

Action:

- Define devices with above 16 megabyte UCBs.

Predicted results:

- Constraint on common storage below 16 megabytes is relieved and private storage expanded to the next megabyte so it is not constrained. There is room for the captured UCBs that the applications need.

Case 3: Some of your applications require 24-bit addresses for UCBs and your common storage is constrained. You cannot define enough above 16 megabyte UCBs to cause the private storage area to expand into the next megabyte and private storage is not currently constrained. Other applications in your installation might have enough room for the captured UCBs that they require or might not require captured UCBs.

Action:

- Be cautious about defining any above 16 megabyte UCBs. Capturing UCBs might cause your private storage to become constrained. There must be enough room in your private storage for captured UCBs for the application.

Predicted results:

- Some constraint on common storage below 16 megabytes is relieved.
- If you limited the number of above 16 megabyte UCBs, your private storage is not constrained. You might choose to move the UCBs for certain applications because there is room for the captured UCBs that they require. When trying to make this decision, remember that the same UCBs might be accessed by more than one application.

Case 4: Some of your applications require 24-bit addresses for UCBs and your common storage is constrained. You cannot define enough above 16 megabyte UCBs to cause the private storage area to expand into the next megabyte and private storage is constrained.

Action:

1. Try to relieve private storage constraint by moving as much data as possible to above 16 megabyte private storage.

   For example, the BSAM, QSAM, BPAM, and BDAM access methods allow data areas to be located above 16 megabytes. This support includes allowing the caller to issue most SAM, PAM, and BDAM macros in 31-bit addressing mode regardless of whether the data is above or below 16 megabytes. See *z/OS DFSMS Macro Instructions for Data Sets* for more information.

Also, if you are using VSAM to allocate devices, you can use the NOCAPTURE option when you dynamically allocate devices so UCBs are not captured.

2. Try to have private storage expand to the next megabyte by changing the CSA and SQA initialization parameters.

   The system rounds the amount of common storage to the next megabyte to create a megabyte boundary between the below 16 megabyte common and private storage areas. You request common storage with the SQA and CSA parameters in the IEASYSxx parmlib member.

   Determine the amount to reduce the CSA/SQA parameter values to force private storage to the next megabyte using the following calculation:

   ```
   (1MB - (actual CSA - requested CSA)) - storage savings
   ```

   actual CSA - requested CSA
   The amount of rounded storage that was allocated to create a megabyte boundary. Determine this amount by subtracting the amount of CSA requested with the CSA parameter in parmlib from the actual amount of CSA present in the system after IPL. You can determine the amount of actual CSA by viewing the RMF Monitor III common storage remaining table.

   storage savings
   The amount of common storage saved by moving UCBs above 16 megabytes. If you have already moved the UCBs, you can look at the actual storage numbers before and after moving UCBs to determine the savings or you might choose to estimate the amount you will save and then make any changes as necessary. The size of a UCB is dependent on the device type. For example, a DASD UCB defined below 16 megabytes has approximately 128 bytes below 16 megabytes. The size of the prefix extension is not included because it already exists above 16 megabytes.

   Remember that you need to reserve SQA/CSA storage for the current workload as well as for future expansion. Make sure that you include that estimate when you request common storage below 16 megabytes with the CSA and SQA parameters. For more information about the parameters, see *z/OS MVS Initialization and Tuning Reference*.

3. Continue to define below 16 megabyte UCBs. Do not define any above 16 megabyte UCBs.

Predicted results:

One of the following:

1. You were able to reduce private storage constraints enough to capture the UCBs as necessary for the application.
2. You were able to tune the CSA and SQA parameters to expand to the next megabyte.
3. You decided not to define above 16 megabyte UCBs for this application.

Case 5: You are adding new devices to your installation, for example, a string of 3390 devices, and you are concerned about below 16 megabyte common and private storage constraints.

Action:

- Define as many UCBs above 16 megabytes as possible to prevent below 16 megabyte common or private storage from becoming constrained.

Define enough above 16 megabyte UCBs to prevent below 16 megabyte common storage from expanding into the next megabyte segment when the IODF with the additional device definitions is used at the next IPL. For example, if only a small amount of below 16 megabyte common storage is used to round CSA to the next megabyte boundary, additional UCBs might cause common storage to expand into the next megabyte. This situation could be a problem if your private storage is constrained because of the expansion of common storage into what otherwise would be private storage. Prevent this problem by defining above 16 megabyte UCBs; however, be aware that some private storage might be used for captured UCBs depending on your applications.

Predicted results:

- Constraint on common below 16 megabyte storage is not adversely affected by the addition of the new devices.

Defining an above 16 megabyte UCB

You can use HCD to define a device's UCB above 16 megabytes if the device's unit information module (UIM) indicates that the device support code enables above 16 megabyte UCBs. If the device support code supports above 16 megabyte UCBs, the LOCANY parameter appears on the HCD Define Device Parameters / Features panel. Specify a value for the LOCANY device parameter to either:

- Define a device's UCB below 16 megabytes only (LOCANY is No). The UCB cannot reside in 31-bit storage.
- Define a device's UCB above 16 megabytes (LOCANY is Yes). The UCB can reside in 31-bit storage.

The default for LOCANY (if it is not specified by the UIM) is No. This can be overwritten by including the option OS-PARM-DEFAULT = LOCANY,YES in the HCD profile.

You can dynamically change the definition of a UCB. To dynamically change a definition from below 16 megabytes to above 16 megabytes, change the LOCANY parameter value from "No" to "Yes" and activate the new IODF.

Defining allocation preferences

This topic gives background information about I/O device allocation, which you need to understand before defining the EDT and esoteric device groups in your configuration.

You can control the allocation preferences for I/O devices using HCD. First, you must understand how devices are allocated for jobs in z/OS.

Understanding I/O device allocation in z/OS

When you submit a job, you identify I/O devices required by the job. The device information can be obtained from either a catalog, SMS overrides, or specific UNIT parameters on DD statements. Before the job can continue execution, z/OS must allocate all those devices to the job.

There are three ways to specify device allocation for a job using UNIT parameters on DD statements. On the UNIT= parameter, specify one of the following:

- A specific device number
- A generic device type
- An esoteric device group

Indicating a specific device for allocation

To request a device explicitly for a job, specify a device number on the UNIT= parameter or on the corresponding dynamic allocation parameter. If that device is available, z/OS allocates the device to the job. However, if the device is unavailable (for example, allocated to another job), your job must wait until the device is available for allocation.

Specifying a generic device type for allocation

z/OS logically groups device types with similar characteristics and assigns the group a generic name. Such a group is called a generic device type. z/OS, for example, groups the 3330-1, 3330-2, and 3333-1 into a generic device type named 3330. Any time a program allocates a 3330, z/OS interprets it to mean any of the devices in that generic device type.

To request a device allocation, you can specify a generic device type on the UNIT= parameter. z/OS allocates a device from the specified generic device type. For example, if you code the following DD statement:

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

z/OS allocates a device from generic device type 3330.

If you specify a device number, such as 3330, that may be mistaken for a unit device type, precede the device number with a slash.

For example, if you code the following DD statement:

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

z/OS allocates the device that has the device number 3330.

Note: An esoteric device name must not contain a slash (/).

For more information, see the UNIT= parameter description in *z/OS MVS JCL Reference*.

Specifying an esoteric device group for allocation

A job that specifies an *esoteric device group* is requesting z/OS to allocate a device from that group. An esoteric device group can include devices of different generic device types. All of the devices that you assign to an esoteric device group must be of the same device class with the following exception: you can define esoteric device groups that contain devices from both DASD and tape device classes but this is not recommended.

Devices belong to one of the following classes:
- Channel-to-channel adapters
- Communication devices
- Direct access devices
- Display devices
- Character readers
- Tape devices
- Unit record devices

To request device allocation, you can specify an esoteric device group on the UNIT= parameter. For instance, DASD and TAPE are group names that identify a group of devices eligible for allocation to satisfy a UNIT=group request (for example, UNIT=DASD, or UNIT=TAPE on the DD JCL statement).

z/OS uses the eligible device table (EDT) to determine the set of devices associated with the esoteric device group. Figure 12 gives an example of relationships among generic and esoteric device groups.

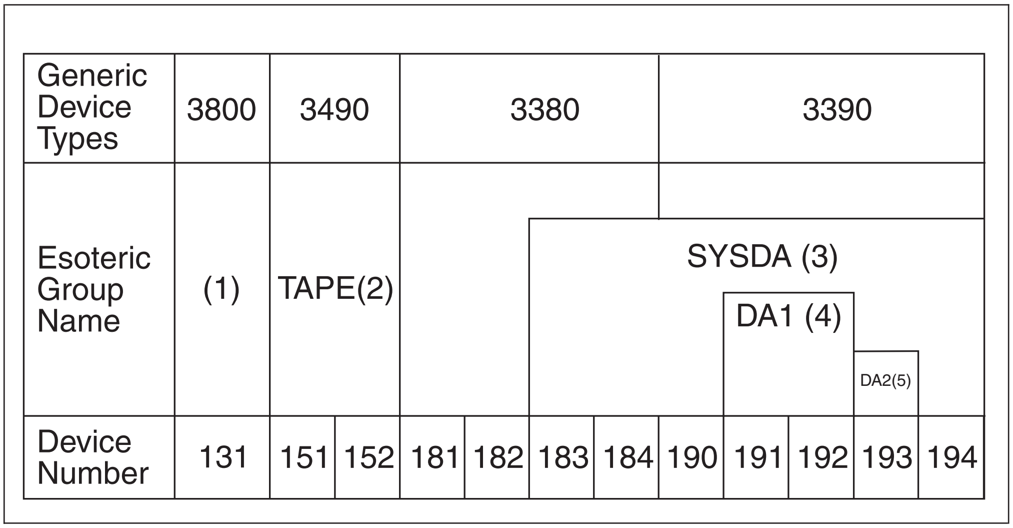


Figure 12. Relationships among Generic and Esoteric Device Groups

Notes:

1. The absence of an esoteric group name means that you can request the 3800 as UNIT=3800, or specify the specific device number, 131.
2. TAPE is the esoteric group name for the two 3490 tape drives, with device numbers 151 and 152.
3. SYSDA is the esoteric group name for two 3380s (device numbers 183 and 184) and five 3390s (device numbers 190-194). When UNIT=SYSDA appears on a DD statement, units 183, 184, 190, 191, 192, 193, and 194 are eligible devices.
4. DA1 is the esoteric group name for two 3390 DASDs (device numbers 191 and 192).
5. DA2 is the esoteric group name for a single 3390 DASD (device number 193). When DA2 is coded on a DD UNIT parameter, device number 193 is the only device eligible for that allocation request.

When z/OS allocates a tape device to a job, it serializes access to the specified esoteric device group until the allocation request is satisfied. Only the devices that belong to the specified esoteric device group are unavailable for allocation to other jobs. When the tape device is automatically switchable, all other automatically switchable devices of that generic device type are unavailable for allocation to other jobs during device selection. Referring to Figure 12, assume that TAPE151 is device 151 and TAPE152 is device 152 and that both devices are automatically switchable. If one jobstep requests UNIT=TAPE151 and during the selection process another jobstep requests UNIT=TAPE152, the system does not consider TAPE152 to be eligible for the second jobstep. This action occurs because both TAPE151 and TAPE152 are generic type 3490 and are automatically switchable.

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 HCD.

The group named SYSSQ must include only tape devices or only direct access devices. The group named SYSDA must include only direct access devices.

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, check the UNIT= parameter on the cataloged procedure DD statements.

Specifying order in an esoteric with the device preference table

An esoteric device group can include devices of more than one generic device type. When z/OS attempts to allocate a demountable device, a unit record, or a printer from an esoteric device group, it turns to the device preference table that defines the order in which to attempt the allocation. The system uses the table to serialize its selection of those devices.

HCD provides a predefined device preference table that defines the generic device types and lists them in order of performance. (The first device type on the list is the fastest. The last device type on the list is the slowest.) You can use the default device preference table values for a generic or you can specify a different value to change the allocation order.

For tape requests, the system considers many factors: the type of request, the devices eligible to satisfy the request, and the characteristics of each eligible device. Based on these factors, the system selects the optimal device to allocate. z/OS allows users to participate in the selection process through the use of SSI function code 78. For details on how to use function code 78, see *z/OS MVS Using the Subsystem Interface*.

Related information:

- For the most current information about device support, see the specific device information or use the “Query supported hardware and installed UIMs” option on the HCD primary task selection panel.
- For information about printing reports through HCD and an example of the Supported Hardware report, see *z/OS HCD User's Guide*.

Advantages of specifying an esoteric device group

By requesting a specific device (for example, UNIT=570), you limit z/OS to a choice of only one device, thereby decreasing the probability that the device will be available when the job needs it. In contrast, by requesting a generic device type, you can allow z/OS a choice of several devices, thereby increasing the probability that a device will be available when the job needs it. Of course, this increase in probability assumes that there will be more than one device of that generic type available for allocation.

When you request a generic device type, z/OS serializes access to that generic device type. This means that z/OS does not allocate devices from that generic device type to other jobs until z/OS satisfies your jobstep allocation request. Because z/OS makes the entire generic device type unavailable for allocation to other jobs, there might be a delay in satisfying other jobs' allocation requests. Therefore, requesting a generic device type increases the probability that there will be a delay in satisfying allocation requests for other jobs. However, waiting for a generic device type generally takes much less time than waiting for a specific device.

Assuming that the esoteric device group identifies more than one device, a device will more likely be available for a UNIT=TAPE request, for example, 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 while z/OS serializes access to the esoteric device group.

To summarize how the system handles the different UNIT specifications, here are three examples. The first two examples describes dedicated devices and the third describes automatically switchable devices.

The first example compares the likelihood of the system being able to satisfy the following specific and generic device requests:
- UNIT=270
- UNIT=3480, where the devices are defined at addresses 270, 273, 285, 294, 366, and 414.

In this case, it is more likely the request can be satisfied by any of the six devices eligible with the UNIT=3480 request than with the UNIT=270 request.

The second example shows how you can subdivide devices into two esoteric specifications so that two jobs can have similar device requests satisfied at the same time. The devices are defined as follows:
- TAPE1 is 270, 273, and 285
- TAPE2 is 294, 366, and 414.

In this case, jobs that request TAPE1 can be in allocation at the same time as jobs that request TAPE2.

The third example is the same as the second example and the UNIT specifications are the same; however, all devices are automatically switchable. In this case, jobs that request TAPE1 and jobs that request TAPE2 cannot be in allocation at the same time. The second job will have to wait for the first job to complete allocation. This will be true whether the jobs are on the same system or on different systems.

Holding or releasing resources while a job waits

Assuming that you are not using JES3 pre-execution setup, to make best use of device allocation, you must decide whether the system should hold or release resources (non-sharable devices and volumes) while a job waits for those resources. When the operator requests that allocation for a specific job is to wait for resources, the system issues a message that asks the operator to reply with either HOLD or NOHOLD.

HOLD

The system holds non-sharable devices and volumes already allocated to the job. Select this option if the needed resources are constantly being freed, and allocation requests for other jobs will probably not be held up by the requests made for this job. This job can hold up other requests in either of two ways:

1. The job has already allocated units needed for another job.
2. The job is waiting for devices to which z/OS is controlling access through serialization.

NOHOLD

The system releases non-sharable devices and volumes already allocated to the job. Select this option if the needed resources may not be freed for some time, and allocation requests for this job are likely to hold up allocation requests issued for other jobs.

Requests for dynamic allocation are not held up by requests waiting for batch allocation, even though the jobs awaiting batch allocation are holding resources.

A dynamic change can be held up from completing if allocation is waiting for a response. For information about automating responses to allocation messages, see *z/OS MVS Installation Exits*.

Device preference table's effect on UNIT=AFF request: When a job requests unit affinity, the order of devices in the device preference table can affect allocation. (See Figure 13.) The lower a device's preference value, the higher its priority. Device preference values affect 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, z/OS allocates both DDs to the same physical unit. That unit:

- Will belong to the specified esoteric device group
- Will have the lowest device preference value from that esoteric device group to appear in the device preference table

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 with the following device preference values:

Types	Values
3480	1100
3400-5	1200
3400-3	1210

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

Figure 13. Example of Device Preference Table Affecting UNIT=AFF

Device preference values also affect the unit-affinity-ignored unit name, that is, the UNITAFF parameter in the ALLOCxx parmlib member. If the installation does not specify UNITAFF, the system default is the tape generic highest in the device preference table. For more information, see the ALLOCxx UNITAFF parameter in *z/OS MVS Initialization and Tuning Reference*.

See Figure 14 on page 59 for an example.

Assumptions

- UNITAFF was not specified in ALLOCxx
- The SMS ACS routines are defined so that:
  - Data set L is to be redirected from tape to an SMS-managed DASD volume, SD3
  - Data set M is not to be redirected and is, therefore, still intended to go to a non-SMS-managed tape volume
- Your installation is using the IBM-provided device preference table. In that table the affected device types appear with the following device preference values:

Types	Values
3490	1000
3480	1100

- Assume the following JCL statements:

```
//JOB2   JOB   ......
//STEP1  EXEC  ......
//DD11   DD     DSN=L,DISP=(NEW),UNIT=3480,.....
//STEP2  EXEC  ......
//DD21   DD     DSN=L,DISP=OLD,......
//DD22   DD     DSN=M,DISP=(NEW,CATLG),UNIT=AFF=DD21
```

Explanation

- In STEP1, DD11, data set L is created and cataloged on SD3, SMS-managed DASD (redirected using SMS ACS routines).
- In STEP2, DD21, data set L is an existing data set, cataloged on SD3, SMS-managed DASD. DD21 is both the referenced DD (referenced by the UNIT=AFF on DD22) and the primary DD.
- In STEP2, DD22 is the referencing DD, which requests unit affinity to DD21. Because data set L is on SMS-managed DASD, the system cannot honor the unit affinity for DD22, which is intended to go to tape. With the unit affinity ignored, the system must determine a unit to be used for DD2. Because UNITAFF was not specified in ALLOCxx, UNITAFF will default to 3490 because it is the tape generic highest in the device preference table. DD22 will be allocated to a 3490.

Figure 14. Example of Device Preference Table Affecting the Default Unit-Affinity-Ignored Unit Name

Note: In a JES3 environment, the IBM-supplied device preference table defines 3480X devices with a higher priority than 3480 devices. Do not alter this priority, because, if you do, the system does not honor a request for affinity between the two device types. A job that requests data set affinity between 3480X and 3480 devices will fail when the installation has specified a higher preference for the 3480 devices.

Defining device allocation in HCD

You use HCD to define esoteric device groups and specify the device preference values of generics within esoteric device groups. You do this by defining eligible device table (EDT) information using HCD.

Information about device allocation is collected in an EDT for z/OS. 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, or during a dynamic configuration change, your installation identifies the EDT that z/OS is to use. During an IPL, you identify the EDT in the LOADxx member. See Chapter 3, "Specifying an I/O configuration at IPL," on page 73 for a description of the LOADxx member. During a dynamic configuration change, you identify the EDT on the HCD Activate New Configuration panel or on the ACTIVATE command. Jobs that run after the IPL or dynamic change can request device allocations from any of the esoteric device groups assigned to the selected EDT.

Your installation can define multiple EDTs in each I/O definition file (IODF). 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

You define each EDT with a unique identifier in HCD.

Note: In a JES3 environment, 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.

Each EDT definition can include:
- Definitions for one or more esoteric device groups (See "Defining esoteric device groups in an EDT using HCD.")
- Changes to the device preference table (See "Changing the device preference table order.")
- Names of the generic device types you want to make eligible for VIO (See "Defining devices as virtual I/O (VIO) eligible" on page 61.)

You can use HCD to view the configuration data in an EDT. Programs can view information from the currently active EDT using the EDTINFO macro. (See *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.)

Defining esoteric device groups in an EDT using HCD

Use HCD to define an esoteric group, assign the group a name, and 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. To request allocation of a device from an esoteric device group, specify the esoteric name on the UNIT= parameter of a JCL DD statement.

In JCL terminology, an esoteric name is called a *group name*.

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.

Do not use esoteric names ***SYSALLDA***, ***SYS3480R***, or ***SYS348XR***, which are reserved for use by IBM.

Changing the device preference table order

The device preference table defines the order of devices that z/OS uses to satisfy an allocation request. To change the device order that z/OS is to use when attempting to satisfy an allocation request, use HCD to define a new preference table order as part of a new EDT definition in the IODF. In the device preference table, devices with lower preference values have higher priority. If the allocation is unsuccessful, z/OS continues the attempt following the order specified in the IBM-provided device preference table.

When you define a device, HCD automatically defines the generic device type in the EDT. You can use HCD to view the generics defined in an EDT. For each generic in an EDT, you can specify an installation device preference table value to specify the order the generic is allocated within an esoteric or you can keep the IBM default device preference value.

Defining devices as virtual I/O (VIO) eligible

VIO refers to data set allocations that exist in paging storage only. z/OS does not use a real device unless z/OS must page out the data set. If z/OS must page out the data set, z/OS 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.

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

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

You can identify one or more *direct access* generic device types that you want to make eligible for VIO.

EDTs for dynamic configuration

Some dynamic configuration changes that you can make to an EDT are:
- Changes to esoteric groups
- Changes to the device preference table values for generics, which determines the order in which devices are allocated from an esoteric group

These dynamic configuration changes could be used to define an IODF with the same hardware configuration but with a different ID for the EDT. The different ID for the EDT allows an installation to activate a configuration with a different device preference order and different esoteric device groups for different jobs, such as off-shift work. Other dynamic configuration changes that affect the EDT are:
- Adding devices, including appropriate new generic names
- Deleting devices
- Modifying devices, which results in deleting then adding the same device numbers (Message IOS502I reports the deleted and added device.)

A dynamic configuration change causes z/OS to activate a new EDT definition.

Primary and secondary EDTs: Usually, z/OS uses one eligible device table (EDT) to process allocation requests. However, when you dynamically change your system's software I/O configuration definition, z/OS must use two EDTs to process the change:

- The **primary EDT** processes all current and new allocation requests. z/OS runs with only the primary EDT until you make a dynamic configuration change. z/OS then activates a new primary EDT for the new I/O configuration, which makes the old primary EDT the secondary EDT.

- The **secondary EDT** receives no new allocation requests. The system deletes the secondary EDT when it finishes the allocation requests issued before the dynamic configuration change.

A second activation may fail if the previous activation is not finished because work is still accessing the secondary EDT.

Migrating to subchannel sets

In a System z environment with z/OS V1R7 or later installed, you can define another subchannel set with ID 1 (SS1) in addition to the existing subchannel set (SS0) in a logical channel subsystem. This ability relieves the constraint for the number of devices that can be accessed by an LPAR. You can define parallel access volume (PAV) alias devices (device types 3380A, 3390A) of the 2105, 2107 and 1750 DASD control units to SS1. In SS0, you can place 63.75K devices, and in SS1 you can place 64K parallel access volume alias devices. Device numbers can be duplicated across subchannel sets. See Figure 5 on page 19.

Scenario: All LPARs are z/OS V1R7 or later

Using subchannel set support when all z/OS LPARs are IPLed on a release that supports subchannel sets is the easier of the two migration methods. Alias devices that are currently in subchannel set 0 (SS0) can be moved into subchannel set 1 (SS1), through your normal dynamic configuration change method. After all changes are made to the IODF and it is active on the System z9 BC or z9 EC, you have successfully reassigned your alias devices from SS0 to SS1. Doing so frees the number of device addresses in SS0 equal to the number of alias devices moved into SS1. Next, define more physical volumes to SS0.

Note that the logical and physical device unit addresses assigned to a device on a control unit must be unique; you cannot define the same CU/device unit address to a device in both SS0 and SS1. For example, if you want to define the PAV base and alias devices in the range 8000-807F in different subchannel sets, but on the same control unit, you can define the devices as follows:

- Base devices, range 8000-807F in SS0, unit address 00-7F (CU number 8000)
- Alias devices, range 8000-807F in SS1, unit address 80-FF (CU number 8000).

Scenario: Some LPARs are pre-z/OS V1R7

When some releases support subchannel sets and some do not, the LPARs that support subchannel sets must be isolated into a separate logical channel subsystem. This is necessary because a device can only be defined to one subchannel set within a logical channel subsystem. When a device is defined to subchannel set 1 (SS1) within a logical channel subsystem, pre-z/OS V1R7 systems that are IPLed into the same logical channel subsystem will not have access to that device.

How to accomplish this depends on whether or not you have an available logical channel subsystem to use for the z/OS V1R7 system. When a logical channel subsystem is available, it has partitions that are not being used.

- One or more logical channel subsystems are available
- All the logical channel subsystems are in use

If there is an available logical channel subsystem, it will be an ideal staging ground to introduce subchannel sets. When the processor is in use prior to z/OS V1R7, an effective migration path with an available logical channel subsystem is as follows:

1. Define a z/OS LPAR into an available channel subsystem

2. Span channels to the new LPAR in the IODF
3. Write the IOCDS out to the processor
4. Dynamically activate the new IODF if all releases are at z/OS V1R6; releases prior to z/OS V1R6 require a POR.
5. Activate the LPAR and IPL z/OS V1R7
6. Convert the current IODF to IODF version 5
7. Define the desired subchannel set size for every logical channel subsystems
8. Create an IODF with a second OS configuration as a copy of the first
9. Move the targeted alias devices from SS0 to SS1 in the second OS configuration.
10. Write the IOCDS out to the CP
11. Update the LOADxx member to specify the new OSCONFIG
12. POR with the new IOCDS.

If no channel subsystem is available, use these actions starting with Step 5.

For more information about defining multiple subchannel sets using HCD, see *z/OS HCD User's Guide*.

Example of migrating devices between subchannel sets

This topic shows a possible approach for organizing your PAV base and alias devices in two different subchannel sets:

- "Numbering schemes" on page 64 introduces some methods for organizing device numbers and unit addresses of the devices.
- "Using HCD to move alias devices into a different subchannel set and closing the gaps in the device number range" on page 67 shows how to use HCD to move PAV alias devices from SS0 to SS1, applying one of discussed numbering schemes to fill the numbering gaps.

To use the information in this section, you must understand PAV and HyperPAV technology, and how to select an appropriate ratio between base and alias devices. Also be aware that the described consolidation of device ranges does not consider any involved migration of data. For this purpose, various approaches are available, such as DFSMSdss, TDMF®, or LDMF.

Further, an understanding of the following topics would be helpful in migrating devices between subchannel sets:
- Availability of free device ranges within the existing 64K channel subsystem limit
- DASD capacity planning detail for a period of 5 years or longer
- Logical Subsystem alignment of existing and future disk subsystems
- Disk subsystem support of new technology (for example, HyperPAV)
- Performance profile of disk subsystems
- DASD replication requirements.

Device number alignment

No correspondence is needed between device numbers in the two subchannel sets. For example, device number 8000 in subchannel set 0 (SS0) and device number 8000 in subchannel set 1 (SS1) can be completely separate devices.

Typically, you define base devices (for example, device type 3390B) from the start of the unit address range (unit address 00 forwards) and define alias devices from the last device in the range backwards. If the available unit address range is completely filled, this becomes unit address FF backwards.

Unit addresses of base and alias devices on a single control unit must be unique; they cannot be duplicated across subchannel sets. Also, base devices and their related alias devices must be defined on the same control unit.

Numbering schemes

Several numbering schemes are possible, each of which is suitable for specific circumstances. Thus, the approach you select depends on your environment and requirements. Also, different implementations of PAV devices might restrict the alignment of alias addresses within the range. Check this with your disk storage vendor before selecting a numbering scheme.

This section introduces the following numbering schemes:
- "Compacting or rolling reuse of addresses"
- "Pairing addresses" on page 65
- "Filling the device number ranges" on page 66.

Compacting or rolling reuse of addresses: You can use SS0 and SS1 to have PAV base and aliases in different subchannel sets, but with the same or correlated device numbers. For example, you can use two aliases for one base device, for example, to ensure that an alias device with a number of 1080 or greater can only be correlated with a base device with a number of 1080 or greater.

This section presents three examples with different ratios between base and alias devices. For example, a ratio of 1:2 means that two alias devices are related to one base device.

Example with ratio 1:2

Control unit	Devices / SSID	Device addresses / UAs
1000	64 3390Bs in SS0	1000-103F / 00 - 3F
	128 3390As in SS1	1000-107F / 40 - BF
1080	64 3390Bs in SS0	1080-10BF / 00 - 3F
	128 3390As in SS1	1080-10FF / 40 - BF
1100	64 3390Bs in SS0	1100-113F / 00 - 3F
	128 3390As in SS1	1100-117F / 40 - BF

With this approach and the 1:2 ratio, you have a range of 64 unit addresses from C0 - FF on each control unit and you have gaps in the device numbering in SS0 between 1040 and 107F, 10C0 and 10FF, and so on. You use the complete capacity of SS1 for defining alias devices, which results in 256 x 64 available UAs spread over 256 control units and also results in the same amount of available device numbers in SS0, which you can use for non-PAV devices or non-DASD devices.

As an alternative, you can use the complete unit address range on a control unit, if you select a ratio of 1:1 or 1:3. The 1:1 ratio would yield a direct correspondence between base and alias device numbers with the unit addresses of the bases in the first half of the range (00-7F) and the unit addresses of the aliases in the second half (80-FF).

Example with ratio 1:1

Control unit	Devices / SSID	Device addresses / UAs
1000	128 3390Bs in SS0	1000-107F / 00 - 7F
	128 3390As in SS1	1000-107F / 80 - FF
1080	128 3390Bs in SS0	1080-10FF / 00 - 7F
	128 3390As in SS1	1080-10FF / 80 - FF
1100	128 3390Bs in SS0	1100-117F / 00 - 7F
	128 3390As in SS1	1100-117F / 80 - FF

Example with ratio 1:3

Control unit	Devices / SSID	Device addresses / UAs
1000	64 3390Bs in SS0	1000-103F / 00 - 3F
	192 3390As in SS1	1000-10BF / 40 - FF
1040	64 3390Bs in SS0	1040-107F / 00 - 3F
	192 3390As in SS1	10C0-117F / 40 - FF
1080	64 3390Bs in SS0	1080-10BF / 00 - 3F
	192 3390As in SS1	1180-113F / 40 - FF

With this approach and the 1:1 ratio, you achieve a complete and consecutive use of the available device number ranges in SS0 and SS1. With the 1:3 ratio, exploitation of device numbers is consecutive, too, in both subchannel sets, but when SS0 is one-third full, SS1 is already used.

Pairing addresses: In this section, we examine the method of pairing addresses, with consideration for different ratios between base and alias devices.

The first half of a control unit's range of unit addresses (128 devices) is allocated to the initial base range. In contrast to the method described in “Compacting or rolling reuse of addresses” on page 64, we then reuse the second half of this device range for the next consecutive assignment of devices within SS1, thus creating a different correspondence between bases and aliases. This method works only when no more than 128 alias devices are defined for a range of base devices. Here, devices are aligned to unit addresses.

Example 1 with ratio 1:1

Range	Control unit	Device addresses / UAs (128 base devices in SS0)	Device addresses / UAs (128 alias devices in SS1)
8000-80FF	8000	8000-807F / 00-7F	8080-80FF / 80-FF
	8080	8080-80FF / 00-7F	8000-807F / 80-FF
8100-81FF	8100	8100-817F / 00-7F	8180-81FF / 80-FF
	8180	8180-81FF / 00-7F	8100-817F / 80-FF

It is now possible to have the following correspondence between base and alias devices, for example, on control units 8000 and 8080:

```
CU 8000:
base devices (SS0) 8000-807F are related to alias devices (SS1) 8080-80FF

and vice versa on CU 8080:
base devices (SS0) 8080-80FF are related to alias devices (SS1) 8000-807F
```

As in “Compacting or rolling reuse of addresses” on page 64, with this approach and the 1:1 ratio, you achieve a complete and consecutive use of the available device number ranges in SS0 and SS1.

Example 2 with ratio 1:2

Range	Control unit	Device addresses / UAs (64 base devices in SS0)	Device addresses / UAs (128 alias devices in SS1)
8000-80FF	8000	8000-803F / 00-3F	8080-80FF / 80-FF
	8080	8080-80BF / 00-3F	8000-807F / 80-FF
8100-81FF	8100	8100-813F / 00-3F	8180-81FF / 80-FF
	8180	8180-81BF / 00-3F	8100-817F / 80-FF

Achieved consecutive device numbering with this approach (ratio 1:2):

```
SS0:                          SS1:
---------------               -----------------

8000-803F ... 8080-80BF       8000-81FF
8100-813F ... 8180-81BF
```

Let's now look at an example with ratio 1:4. Here, we see that we still achieve full ranges for the alias devices in SS1, however, the gaps in SS0 for the base devices are increasing, compared to ratio 1:2.

Note: A ratio of 1:3 (32 bases/96 aliases) also yields gaps in SS1 (in addition to the gaps in SS0).

Example 3 with ratio 1:4

Range	Control unit	Device addresses / UAs (32 base devices in SS0)	Device addresses / UAs (128 alias devices in SS1)
8000-80FF	8000	8000-801F / 00-1F	8080-80FF / 80-FF
	8080	8080-809F / 00-1F	8000-807F / 80-FF
8100-81FF	8100	8100-811F / 00-1F	8180-81FF / 80-FF
	8180	8180-819F / 00-1F	8100-817F / 80-FF

Achieved consecutive device numbering with this approach (ratio 1:4):

```
SS0:                           SS1:
---------------                ----------------

8000-801F ... 8080-809F        8000-81FF
8100-811F ... 8180-819F
```

Filling the device number ranges: This approach preserves the ranges in SS0 and aligns devices to unit addresses. It is driven by the desired base to alias ratio that was calculated in advance (potentially through the PAV analysis tool).

Basic principles used here are:

1. Establish base to alias ratio first. Align to boundaries of 16 devices.
2. Stick to this alignment for all device ranges.
3. All ranges are filled with 256 devices (00-FF).
4. Use up all device ranges.
5. Define alias devices to SS1.
6. Reuse device numbers used in SS1 as base device numbers in SS0.
7. Reuse device numbers allocated in SS0 in SS1.

Example: 192 bases, 64 aliases (ratio 3:1)

- 256 available ranges exist for each of 256 devices (256 x 256 = 65536 or FFFF(hex) devices).
- 64 alias devices in SS1 free 256 x 64 = 16384 devices in SS0.
- 16384 / 192 = 85 additional ranges for defining base devices in SS0.

Using this example, you can create a worksheet to determine the available ranges.

Table 5. Worksheet showing available ranges

Range	Control unit	192 base devices used in SS0	Free range for 64 base devices in SS0	64 alias devices used in SS1	Free range for 192 alias devices in SS1
1000-10FF	1000	1000-10BF (UA 00-BF)	10C0-10FF	10C0-10FF (UA C0-FF)	1000-10BF
1100-11FF	1100	1100-11BF	11C0-11FF	11C0-11FF	1100-11BF
1200-12FF	1200	1200-12BF	12C0-12FF	12C0-12FF	1200-12BF

Table 5 shows the available device numbers in SS0. Remember, there is no correspondence between the device number and the unit address. Thus, in a single new control unit, you can define the “spare” device numbers accordingly:

Example: new CNTLUNIT 2000

```
Available device numbers in SS0:

64 devices: 10C0-10FF (UA 00-3F)
64 devices: 11C0-11FF (UA 40-7F)
64 devices: 12C0-12FF (UA 80-BF)
--------------------------------
===> a total of 192 free devices numbers, however discontinuous

Available device numbers in SS1:

64 devices to be defined on CNTLUNIT 2000: 1000-103F (UA C0-FF)
```

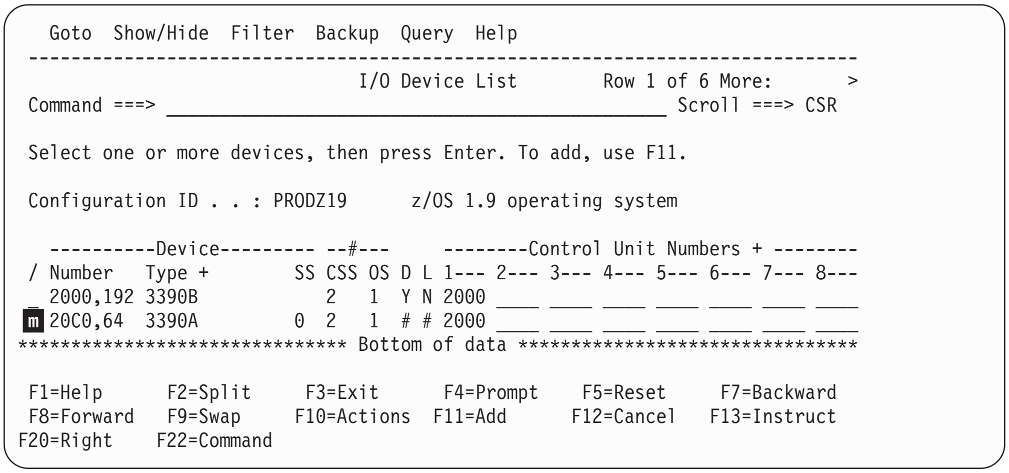
This action creates a range of 4 * 64 = 256 unused device numbers, distributed over subchannel sets SS0 and SS1 as shown in the worksheet. A further 192-64=128 device numbers are available in SS1 to be defined on a different control unit.

Using HCD to move alias devices into a different subchannel set and closing the gaps in the device number range

This section uses the approach described in “Filling the device number ranges” on page 66. Ideally, the end result is a relationship between device numbers of base and aliases in the subchannel sets, and the freed device numbers in subchannel set 0 can be reused without disturbing existing numbering schemes.

Assume that you have the following configuration, in which all devices defined to subchannel set 0 are related to all channel subsystems of a processor:

- 192 base devices 2000 to 20BF of type 3390B are connected to control unit 2000 (type 2107), with unit addresses 00 through BF.
- 64 alias devices 20C0 to 20FF of type 3390A are connected to the same CU 2000, having the unit addresses C0 through FF.



```
  Goto  Show/Hide  Filter  Backup  Query  Help
 ------------------------------------------------------------------------------
                               I/O Device List       Row 1 of 6 More:       >
 Command ===> _______________________________________________ Scroll ===> CSR

 Select one or more devices, then press Enter. To add, use F11.

 Configuration ID . . : PRODZ19      z/OS 1.9 operating system

   ----------Device--------- --#---     --------Control Unit Numbers + --------
 / Number   Type +        SS CSS OS D L 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
 _ 2000,192 3390B            2   1  Y N 2000 ____ ____ ____ ____ ____ ____ ____
 m 20C0,64  3390A         0  2   1  # # 2000 ____ ____ ____ ____ ____ ____ ____
******************************* Bottom of data ********************************

 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F11=Add      F12=Cancel   F13=Instruct
F20=Right    F22=Command
```

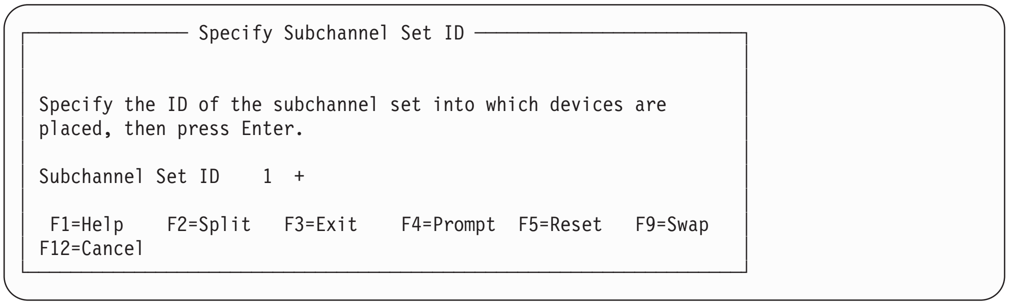
Figure 15. Original device configuration - I/O Device List launched from the related OS configuration

Using the configuration shown in Figure 15, suppose that you want to move the 64 3390A alias devices from subchannel set 0 to subchannel set 1 to gain space for defining new devices. Also, because you have used the numbering scheme in "Filling the device number ranges" on page 66, you have gaps of device numbers in SS0, which you want to fill by redefining the device numbers for the 3390B devices.

You cannot directly change device numbers in the HCD dialog without deleting and recreating the devices. To avoid an awkward and error-prone process, perform the following steps:

1. Create a work IODF from the production IODF to be changed.
2. Navigate to the required 3390A device group on the ***I/O Device List*** (see Figure 15). You can use a filter for all alias devices of type 3390A. Then use action ***Subchannel Set ID group change*** from the context menu (or action code m).

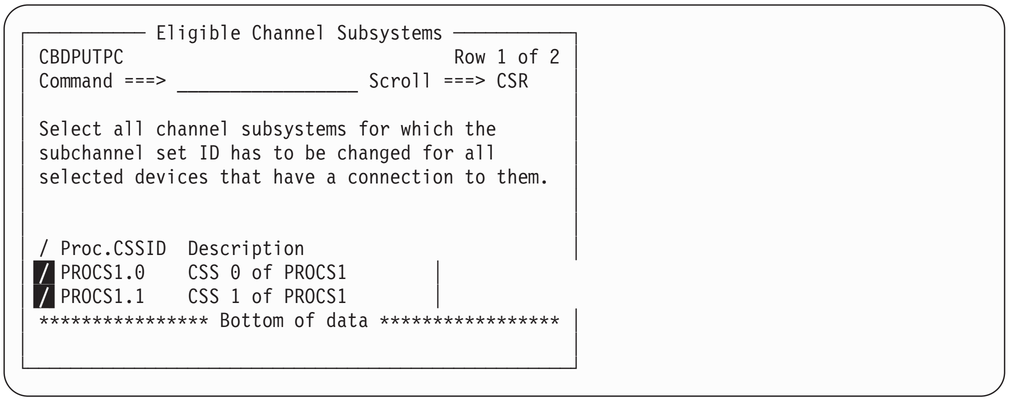
   HCD displays panel ***Specify Subchannel Set ID*** as shown in Figure 16 on page 69. Set the Subchannel Set ID to **1**.



```
Specify Subchannel Set ID

Specify the ID of the subchannel set into which devices are
placed, then press Enter.

Subchannel Set ID    1  +

 F1=Help    F2=Split   F3=Exit    F4=Prompt  F5=Reset   F9=Swap
F12=Cancel
```

Figure 16. Specify Subchannel Set ID

3. Press Enter to display panel ***Eligible Channel Subsystems*** (Figure 17). Select all of the channel subsystems for which you want to change the subchannel set ID.

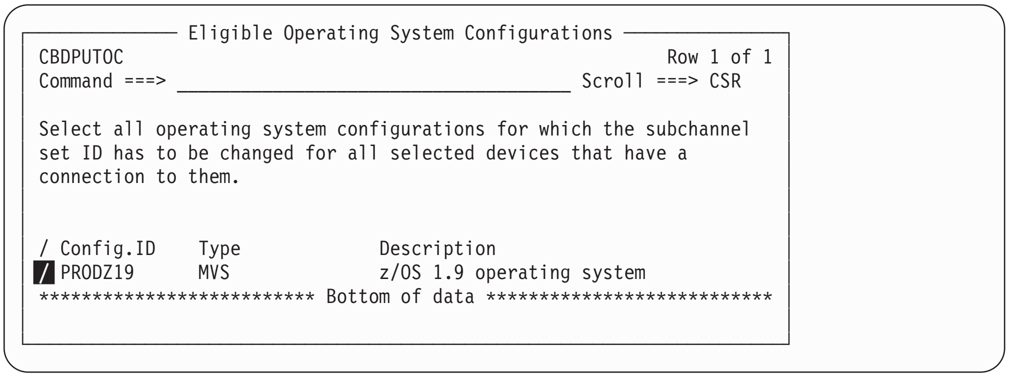


```
Eligible Channel Subsystems
CBDPUTPC                                Row 1 of 2
Command ===> _________________ Scroll ===> CSR

Select all channel subsystems for which the
subchannel set ID has to be changed for all
selected devices that have a connection to them.


/ Proc.CSSID  Description
/ PROCS1.0    CSS 0 of PROCS1
/ PROCS1.1    CSS 1 of PROCS1
**************** Bottom of data *****************
```

Figure 17. Eligible Channel Subsystems

Select both channel subsystems with /.

4. HCD now displays panel ***Eligible Operating System Configurations*** (see Figure 18) where you select the operating systems for which you want to change the subchannel set ID for selected devices.



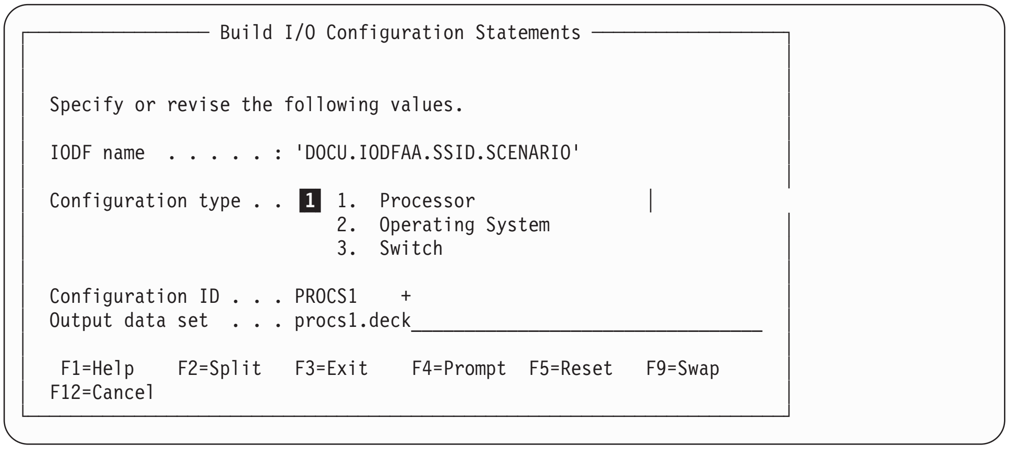
```
Eligible Operating System Configurations
CBDPUTOC                                                    Row 1 of 1
Command ===> _____________________________________ Scroll ===> CSR

Select all operating system configurations for which the subchannel
set ID has to be changed for all selected devices that have a
connection to them.


/ Config.ID    Type             Description
/ PRODZ19      MVS              z/OS 1.9 operating system
************************* Bottom of data ***************************
```

Figure 18. Eligible Operating System Configurations

As a result, the alias devices 20C0 to 20FF remain attached to the same control unit 2000 having the unit addresses C0 through FF, but are moved to SS1.

Now you can change the device numbers of devices 2000 - 20FF to the gaps listed in the scenario from “Filling the device number ranges” on page 66 without deleting and recreating these devices.

5. Make a mapping list of old versus new device numbers. In more complicated scenarios, where a large number of devices attached to different CUs are involved, you might also need to map old CU numbers to new CU numbers. Consider writing an exec program to perform this step.
6. Using the current work IODF, select HCD dialog option 2.10 (***Activate or process configuration data ---> Build I/O configuration statements***) to write the following:
   a. A processor configuration deck for every processor having CU 2000 and the devices 20C0 - 20FF attached that you want to rename (specify **1** as the configuration type as shown in Figure 19).
   b. An OS configuration deck for every operating system with these devices attached (specify **2** as the configuration type).



```
 ---------------- Build I/O Configuration Statements -----------------

  Specify or revise the following values.

  IODF name  . . . . . : 'DOCU.IODFAA.SSID.SCENARIO'

  Configuration type . . 1  1.  Processor
                            2.  Operating System
                            3.  Switch

  Configuration ID . . . PROCS1    +
  Output data set  . . . procs1.deck_________________________________

   F1=Help    F2=Split   F3=Exit    F4=Prompt  F5=Reset   F9=Swap
  F12=Cancel
```

Figure 19. Build I/O Configuration Statements

7. Edit every processor configuration deck created in Step 6a and do the following:
   a. Replace device numbers of alias and base devices in the appropriate IODEVICE statements:

      In our scenario, change from:

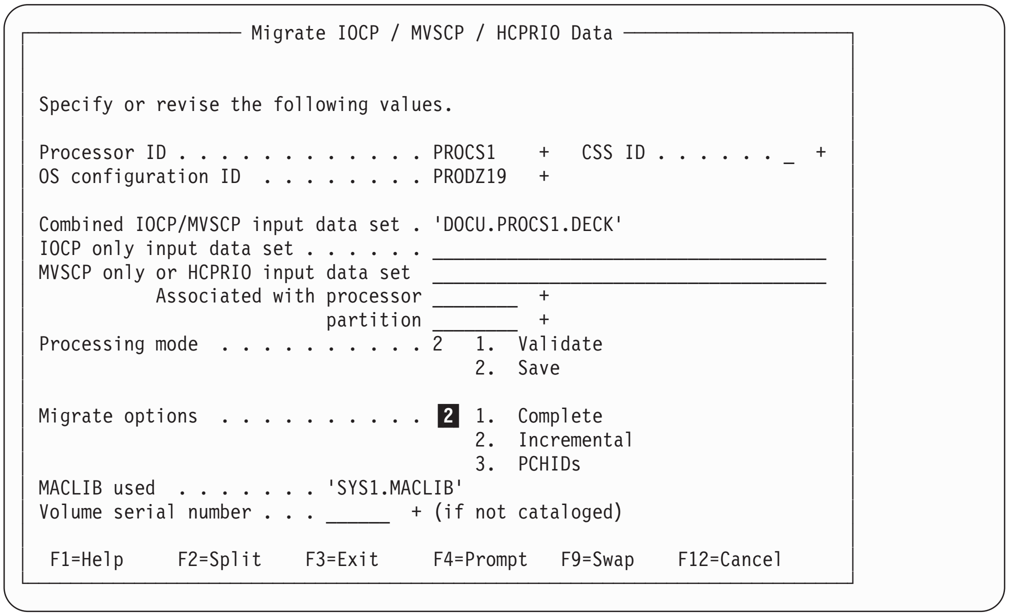
      ```
       IODEVICE ADDRESS=(2000,192),UNITADD=00,CUNUMBR=(2000),        *
             STADET=Y,UNIT=3390B
       IODEVICE ADDRESS=(20C0,64),UNITADD=C0,CUNUMBR=(2000),STADET=Y,*
             SCHSET=1,UNIT=3390A
      ```

      to:

      ```
       IODEVICE ADDRESS=(10C0,64),UNITADD=00,CUNUMBR=(2000),         *
             STADET=Y,UNIT=3390B
       IODEVICE ADDRESS=(11C0,64),UNITADD=40,CUNUMBR=(2000),         *
             STADET=Y,UNIT=3390B
       IODEVICE ADDRESS=(12C0,64),UNITADD=80,CUNUMBR=(2000),         *
             STADET=Y,UNIT=3390B
       IODEVICE ADDRESS=(1000,64),UNITADD=C0,CUNUMBR=(2000),         *
             STADET=Y,SCHSET=1,UNIT=3390A
      ```
   b. For more complex scenarios, you might want to replace CU numbers in the appropriate CNTLUNIT and IODEVICE statements. Alternatively, you can change CU numbers before Step 6 using the ***Change Control Unit Definition*** dialog.

Note: For more complex scenarios, especially if the IODFs are well structured, you might want to write an exec program to automatically perform this step and Step 8.

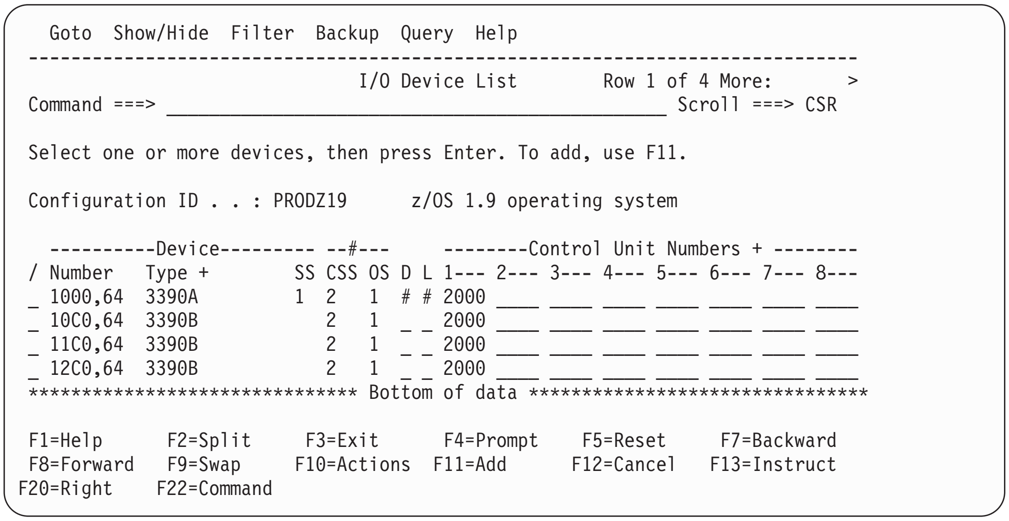
8. Edit every OS configuration deck created in Step 6b on page 70 and do the following:
   a. Replace device addresses of alias and base devices in the appropriate IODEVICE statements. The changes are similar to the changes in Step 7a on page 70.
   b. If required, replace the CU numbers of alias and base devices in the appropriate IODEVICE statements (see also Step 7b on page 70 for an alternative)
   c. If required, replace device addresses of base devices in the appropriate EDT statements.
9. In the work IODF, delete all of the base/alias devices in the device number range to be moved.
10. If required, delete all of the CUs in the CU number range to be moved from the work IODF.
11. Use HCD dialog option 5.1 (***Migrate configuration data ---> Migrate IOCP/OS data***) to do the following:
    a. Migrate the updated processor configuration decks into the work IODF (Migrate option = Incremental (2))
    b. Migrate the updated OS configuration decks into the work IODF (Migrate option = Incremental (2)).



```
                    Migrate IOCP / MVSCP / HCPRIO Data


Specify or revise the following values.

Processor ID . . . . . . . . . . . . PROCS1    +   CSS ID . . . . . . . _  +
OS configuration ID  . . . . . . . . PRODZ19   +

Combined IOCP/MVSCP input data set . 'DOCU.PROCS1.DECK'
IOCP only input data set . . . . . . ____________________________________
MVSCP only or HCPRIO input data set  ____________________________________
           Associated with processor ________   +
                           partition ________   +
Processing mode  . . . . . . . . . . 2   1.  Validate
                                         2.  Save

Migrate options  . . . . . . . . . . 2   1.  Complete
                                         2.  Incremental
                                         3.  PCHIDs
MACLIB used  . . . . . . . 'SYS1.MACLIB'
Volume serial number . . . ______  + (if not cataloged)

 F1=Help     F2=Split    F3=Exit     F4=Prompt   F9=Swap    F12=Cancel
```

Figure 20. Migrate IOCP / MVSCP / HCPRIO Data

12. Review the new I/O Device List to verify the results. In the I/O device list started from the related OS configuration (Figure 21 on page 72), note that the 3390A devices are defined to subchannel set 1. The previously occupied device number range 2000 - 20FF is now available for use on a new control unit.



```
   Goto  Show/Hide  Filter  Backup  Query  Help
 ------------------------------------------------------------------------------
                                I/O Device List        Row 1 of 4 More:       >
 Command ===> _______________________________________________ Scroll ===> CSR

 Select one or more devices, then press Enter. To add, use F11.

 Configuration ID . . : PRODZ19      z/OS 1.9 operating system

   ----------Device--------- --#---     --------Control Unit Numbers + --------
 / Number   Type +        SS CSS OS  D L 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
 _ 1000,64  3390A         1  2   1   # # 2000 ____ ____ ____ ____ ____ ____ ____
 _ 10C0,64  3390B            2   1   _ _ 2000 ____ ____ ____ ____ ____ ____ ____
 _ 11C0,64  3390B            2   1   _ _ 2000 ____ ____ ____ ____ ____ ____ ____
 _ 12C0,64  3390B            2   1   _ _ 2000 ____ ____ ____ ____ ____ ____ ____
 ******************************* Bottom of data *********************************

 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Reset     F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F11=Add      F12=Cancel   F13=Instruct
F20=Right    F22=Command
```

Figure 21. New I/O Device List after migration

Chapter 3. Specifying an I/O configuration at IPL

You must specify your I/O configuration definition at IPL— during initial install, upgrade, and recovery.

Defining your initial I/O configuration

You define your initial I/O configuration in the following situations:

- You could be installing z/OS for the first time. In which case, you would have a bootstrap system, consisting of an install IOCP (stand-alone IOCP) and a bootstrap z/OS. This would require an initial "bootstrap" POR and probably IPL.
- You could be using IODFs for the first time and want to use dynamic configuration capability. In this case, you could run HCD and create the IODF. Do a POR to get the active IOCDS token in the HSA. See “Ensuring that the software and hardware definitions match” on page 6 for information about configuration tokens.

Each time you install a new processor that does not have an IOCDS yet, you can use HCD to build an IOCP input data set. Then you can transfer this data set to tape and run the IOCP stand alone program that is installed in your processor.

Take the following general actions to define your configuration at IPL:

1. Create I/O software configuration plan.

   See Chapter 2, “Planning your I/O configuration definition,” on page 11.
2. Install your I/O devices.

   See device guides for information about devices.
3. Install z/OS, which includes HCD.

   See *z/OS Program Directory* and *z/OS HCD User's Guide*.
4. Install software support for IBM equipment.

   The UIM points to other device support information provided with the product. See program directories for IBM products such as DFSMS or Virtual Telecommunications Access Method (VTAM).
5. Provide and/or install software support for devices not supported by z/OS.

   See *z/OS MVS Device Validation Support*.
6. Determine whether to authorize access to making dynamic configuration changes. The security administrator can use RACF to authorize operators and HCD users to activate dynamic changes. Before anyone can use either the ACTIVATE command or the HCD Activate New Configuration panel, the security administrator must give that user UPDATE authority to the MVS.ACTIVATE resource in the OPERCMDS class. (See *z/OS MVS Planning: Operations*.)
7. To handle data sets that are cataloged with an esoteric name, do one of the following:
   - Specify a token for the esoterics. Allocation uses the esoteric token to find the appropriate esoteric for a data set that has been cataloged using esoterics. You no longer have to keep the esoterics in the same sequence and can delete and add esoterics without having problems accessing data sets that are cataloged using esoterics.

- Recatalog the data sets with a generic device type name (for example, 3380) before using HCD to migrate IOCP/MVSCP data.

8. Invoke HCD.

   You start HCD like any other ISPF application in your TSO/E system. The default TSO/E CLIST that is supplied with HCD is CBDCHCD (which invokes CBDCHCD1) in library SYS1.SCBDCLST.

   See *z/OS HCD User's Guide* for more information, including which functions are available as batch utilities and thus do not require ISPF.

9. Use HCD to define your configuration:

   a. Create a work IODF by either migrating MVSCP/IOCP decks with the HCD panels or migration batch facility or updating an existing IODF.

      Note: If you are not using dynamic capability, sometimes you can anticipate I/O configuration changes. For example, you might know that a control unit and associated DASDs will be installed but are not installed when z/OS is IPLed. Two possible methods of adding the control unit and the DASD are:

      - Define the control unit and DASD to both the hardware and the software before devices are installed. This strategy saves a POR and IPL when the physical control unit and DASD are installed. However, the inactive UCBs waste storage until devices are installed.
      - Define the control unit and DASD only to the hardware before devices are installed. During POR, hardware definitions of the devices are built. When the control unit and DASD are installed, create the software definition, then IPL z/OS with the new software definition to make the devices available. This strategy saves a POR.

   b. Create a production IODF from the work IODF.

   See *z/OS HCD User's Guide* for information about creating IODFs.

   You do not need to pre-allocate an IODF; you allocate it through HCD, and it resides on the volume you specify through HCD.

   You specify an EDT identifier and create an EDT as part of defining your configuration in HCD. You also specify an operating system configuration identifier.

10. Create the LOADxx member in which you specify an IODF statement to identify the I/O configuration definition for IPL.

    An IODF statement contains the name of the production IODF and identifier of the EDT and operating system configuration that is used to IPL the system. The IODF identified becomes the active IODF for the system.

    The Load Parameter Definition (8 byte field) on the SYSCTL or OPRCTL frame on ES/9000 processors tells which LOADxx member to use.

    See "IODF processing at IPL" on page 76 for more information.

11. Specify the configuration to the hardware.

    Use HCD to run IOCP to create I/O configuration data sets (IOCDSs). To create IOCDSs, you can also use IOCP directly without using HCD, but then dynamic capability will not be available. See *IOCP User's Guide*.

12. Unless you are running on a z890, z900, or later processor, you must specify the HSA and MIF expansion requirements for any future dynamic changes at POR. To enable dynamic I/O configuration changes, you must also indicate that you want to allow dynamic modifications. You specify this information in the Activation Reset profiles.

If you expect to make dynamic configuration changes, you must specify a percent of expansion for the size of HSA. (See *Enterprise System/9000 Operator Guide* .) You indicate the percent increase of existing HSA size. A 10% expansion factor will allow 10% more devices and control units. For example, when you have ten devices and want to add 2 devices, you specify 20 percent expansion.

The amount of HSA storage that the channel subsystem requires depends on:

- The configuration definition (the number of devices, control units, and channel paths defined in the IOCDS) used for the POR
- The processor model and microcode level

There is an HSA option on the DISPLAY IOS,CONFIG command that displays the amount of HSA remaining for dynamic changes. For a description, see *z/OS MVS System Commands*.

You can only increase the HSA expansion percentage by performing another POR.

Unless you are running on a z900, z890, or later processor, you also need to specify the expansion factor for MIF. For prior processors, indicate the percentage of the expansion that is for shared devices. For example, if you specify 50 percent, half the expanded HSA will be used for additional shared devices and the other half for additional unshared devices. If you specify 100 percent, all the expanded HSA will be used for additional shared devices. See the processor manuals for more information.

On a z890, z900, or later processor, you do not specify HSA and MIF expansion percentages. Instead, specify the maximum number of devices that you will want to access. When you define each logical channel subsystem, this value replaces the HSA and MIF expansion requirements. This number should take into account any changes you may want to make in the future. You can specify up to 63K devices.

13. Do a POR to initialize the channel subsystem/HSA from an IOCDS created by HCD.

    When you go to make changes to your configuration, tokens in the current IODF and HSA must match in order to make hardware changes. To get the matching token in the HSA, you have to perform a POR with an IOCDS created by HCD. The IOCDS must contain a hardware configuration definition included with the current IODF.

14. When specifying system parameters for IPL, use the SQA parameter in the IEASYSxx member of SYS1.PARMLIB to define estimated additional storage for UCBs that will be added by future dynamic changes. System parameters can be specified either in the IEASYSxx member of SYS1.PARMLIB or as operator responses to IPL messages.

    At IPL the system determines from the current I/O configuration how much SQA and ESQA to use to represent the UCBs for the I/O configuration being IPLed. This amount of SQA and ESQA storage is then increased by the amount that you request on the SQA parameter in the IEASYSxx member of SYS1.PARMLIB. (See *z/OS MVS Initialization and Tuning Reference*.)

    | You must ensure that there is also enough system queue area (SQA) storage for any devices that are to be dynamically added to a configuration. ~~A UCB on average needs 112 (hex) bytes of storage below 16 megabytes and 88 (hex) bytes above 16 megabytes.~~

    When you dynamically add UCBs, the system obtains storage for these new UCBs from the SQA allotment. For example, if you add 10 devices to your configuration, the system obtains about AB4 (hex)/2740 (decimal) bytes of SQA and 550 (hex)/1360 (decimal) bytes of extended system queue (ESQA) for the new UCBs.

Note: If your installation needs to dynamically change between different configurations, in which few or none of the devices are the same, your installation must ensure there is enough SQA for the largest configuration and for the next largest configuration to be used. z/OS needs enough storage for all the UCBs in the current configuration and all the new and changed UCBs in the replacement configuration. If there is a problem during activation of the replacement configuration, z/OS needs this storage to recover and reestablish the UCBs that existed before the attempt to change the configuration.

15. When specifying system parameters for IPL, use the CMB parameter to define estimated additional storage for channel measurement blocks (CMBs) that will be added by future dynamic changes. System parameters can be specified either in the IEASYSxx member of SYS1.PARMLIB or as operator responses to IPL messages.

    A channel measurement block accumulates measurement data for I/O devices. You need to specify, on the CMB parameter in the IEASYSxx member of SYS1.PARMLIB, the sum of the number of I/O devices that you need to measure that are not DASD or tape (or other types of devices specified using the CMB options) plus the number of devices that you plan to dynamically add and measure. (See *z/OS MVS Initialization and Tuning Reference*.)

    To determine the number of devices that the system can measure during a given IPL, the system calculates the sum of the following:
    - Number of DASD devices in the I/O configuration definition that you selected at IPL
    - Number of tape devices in the I/O configuration definition that you selected at IPL
    - Value that you specified on the CMB parameter.

    Therefore, when you define the CMB parameter, remember that the system allocates CMB space for the DASD and tape devices contained in the I/O configuration definition that you select at IPL.

    If you have z890/z990 exploitation support installed on your system, the system ignores the CMB (channel measurement blocks) parameter in the IEASYSxx parmlib member. z/OS dynamically allocates enough space to accommodate the number of devices in use, up to the maximum number of devices possible.
16. Initialize z/OS with the production IODF and select the appropriate LOADxx member for the configuration. See “IODF processing at IPL” for more information.

IODF processing at IPL

When you perform an IPL, the production IODF that defines the configuration to the system is selected and used. This IODF is selected from any number of other IODFs. You identify the IODF by the IODF statement in the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB PDS.

To identify the IODF selected at IPL, do the following:

1. Identify the IODF volume. An IODF volume is the direct access storage device (DASD) on which you place your IODFs. You identify the IODF volume on the LOAD parameter on the operator control (OPRCTL) or system control

(SYSCTL) frame that is used when the system software is loaded or in the LOAD profiles (for a 9672 Parallel Transaction Server). See the processor documentation for more information.

On the LOAD parameter, you specify the device number of the IODF volume and the identifier for the LOADxx member. If you do not specify an IODF volume on the LOAD parameter, the system searches the sysres volume. If you do not specify a LOADxx member on the LOAD parameter, the system uses LOAD00.

The SYSn.IPLPARM or SYS1.PARMLIB PDS contains the LOADxx member. See "Search order for the LOADxx member and IODF" on page 78 for a description of the order in which the PDSs are searched.

2. In the LOADxx member, include an IODF statement that identifies the production IODF for IPL. For details about LOADxx, see *z/OS MVS Initialization and Tuning Reference*.

Figure 22 shows how the IODF is identified for IPL.

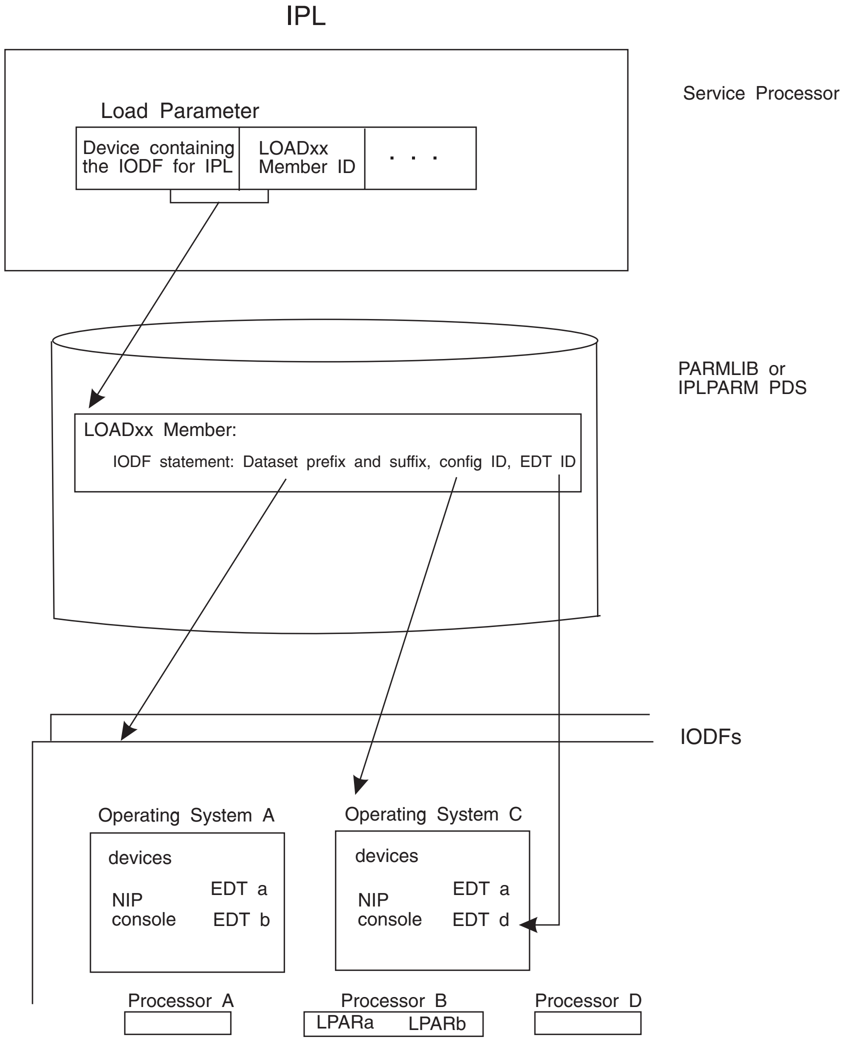


Figure 22. Specifying IODF at IPL

You can have multiple operating system and processor configurations in one IODF. The configurations can share common device definitions. In the IODF shown in the example, the configurations for operating system A and operating system C share EDT a.

IODF statement in the LOADxx member

A production IODF name has the following syntax:

```
<IODF-prefix>.IODF<IODF-suffix>
```

An example of a production IODF name is SYS2.IODF01.

You can specify the IODF prefix and suffix in the LOADxx statement to identify the production IODF at IPL. The IODF statement in the LOADxx member can contain the following information:

- IODF suffix

  The IODF suffix is the two digit hexadecimal number that is part of the IODF name. For example, “01” is the IODF suffix for IODF01.
- IODF prefix

  The IODF prefix is an 8-byte high-level qualifier of the IODF data set name. For example, IBMCONF is the IODF prefix for the IODF data set IBMCONF.IODF01.
- Operating system (OS) configuration identifier

  The 8-byte operating system configuration identifier is part of the configuration definition in the IODF; it identifies the operating system configuration.
- Eligible device table (EDT) identifier

  The EDT identifier is the two digit hexadecimal number that identifies the EDT for use with the configuration defined by the IODF. If you do not specify an EDT identifier, it defaults to “00.”
- LOAD option

  The LOAD option is a field where you indicate if the system will load device support code for every UIM whether a device of that type is installed or not. If you leave the field blank or specify Y, the system loads all the device support modules for the devices in the IODF and all device types that support dynamic configuration for which you have UIMs installed. This option enables you to add device types dynamically that were not in the IODF at IPL. When you specify any other character, the system loads only the modules required for the devices defined in your current IODF.

Search order for the LOADxx member and IODF

During IPL, the system uses the LOADxx member that it finds first when searching in the following order:

1. The system first searches the IODF volume for a SYS**n**.IPLPARM PDS where **n** is 0 through 9.
2. If it does not find SYSn.IPLPARM, it searches the IODF volume for a SYS1.PARMLIB PDS.
3. If it does not find SYS1.PARMLIB on the IODF volume, it searches for SYS1.PARMLIB on the sysres volume.

   Specifying the IODF on the sysres volume is not recommended for the following reasons:
   - It is difficult to replicate the sysres volume if it contains an IODF because an IODF is a VSAM data set.
   - When you share the sysres volume across multiple systems, I/O contention occurs when the systems are IPLed. Having the IODF on the sysres volume adds to the contention.

The LOADxx member may reside on either the IODF volume or the SYSRES volume.

If z/OS finds SYSn.IPLPARM or SYS1.PARMLIB, but it doesn't contain the LOADxx member in the LOAD parameter, z/OS enters a wait state.

Because the system data sets are searched in order, IBM suggests using SYS0.IPLPARM for improved IPL performance.

Specifying the IODF high-level qualifier: If LOADxx is in SYS1.PARMLIB, you must specify an IODF high-level qualifier in the IODF statement. If the LOADxx member is in SYSn.IPLPARM and the IODF statement does not contain a high-level qualifier, z/OS defaults to the high-level qualifier of the SYSn.IPLPARM.PDS.

If you optionally specify equals (========) as the IODF high-level qualifier in the IODF statement, z/OS attempts to use the high-level qualifier from the hardware configuration token in the HSA. If the token is not available, z/OS loads wait state X'0B1', with reason code X'00B'. If the high-level qualifier found in the hardware configuration token contains a blank in the first character, z/OS loads wait state X'0B1', with reason code X'009'.

Specifying the IODF suffix: You can optionally specify an IODF suffix and operating system configuration ID in the IODF statement in LOADxx, or you can let z/OS find the correct IODF automatically. This automatic search saves you from having to change the LOADxx member every time you switch IODFs. However you might have to rename the suffix of the IODF so that z/OS will find it first in its search sequence.

The IODF search options (++, −−, ==, **, and ƀƀ(blank blank)) provide flexibility in IODF management. You can set up LOADxx so that you do not have to change it, following every dynamic configuration and IOCDS update, for use at the next IPL. Users, who want to maintain a back up IODF on a second volume in case of device failure, can use the flexible searching algorithm. This allows the backup IODF name to have the next larger or lower sequential number as a suffix.

Table 6 on page 80 describes the options you can use when identifying the IODF and operating system configuration ID for IPLing the system.

When you specify asterisks, pluses, minuses, or equals as the IODF suffix, z/OS uses the IODF name in the hardware configuration token in the HSA. If you override the descriptor fields (when building a production IODF using HCD panels) and specify an invalid IODF name, the token in the HSA is unusable. In this situation, z/OS searches for a valid production IODF as follows:

- For '**' and '++', z/OS searches suffixes X'00' to X'FF' for a valid IODF
- For '−−', z/OS searches suffixes X'FF' to X'00' for a valid IODF
- For '==', z/OS loads wait state X'0B1', with reason code X'00B'

Attention: If you are specifying asterisks, equals, pluses, or minuses for the IODF suffix in LOADxx, never change the default descriptor field values on the HCD panel when you create a production IODF because z/OS uses this information to find the current IODF. Also, make it a policy to always write a new IOCDS after a dynamic configuration change.

Table 6. Specifying an IODF for IPL

What You Can Specify in the IODF Statement	How z/OS Finds the IODF at IPL	How to make sure z/OS finds a new IODF at IPL
Asterisks '**' for IODF suffix	z/OS uses the IODF that you were running with. If you are IPLing initially or z/OS cannot find a current IODF, z/OS will behave as if you left the IODF suffix blank.	When you activate a different IODF, you do not have to change LOADxx or rename IODF suffixes.
Pluses '++' for IODF suffix	z/OS uses the IODF that you were running with. If you are IPLing initially or there is no current IODF, z/OS behaves as if you left the IODF suffix blank. If the current IODF is not available, z/OS searches for the next available IODF starting with the current IODF suffix number plus one and searches to X'FF'. If none is found, the search continues from X'00' up to the current IODF number.	When you activate a different IODF, you do not have to change LOADxx or rename IODF suffixes.
Minuses '−−' for IODF suffix	z/OS uses the IODF that you were running with. If you are IPLing initially or there is no current IODF, z/OS behaves as if you left the IODF suffix blank. If the current IODF is not available, z/OS searches for the next available IODF starting with the current IODF suffix number minus one and searches to X'00'. If none is found, the search continues from X'FF' down to the current IODF number.	When you activate a different IODF, you do not have to change LOADxx or rename IODF suffixes.
Equals '==' for IODF suffix	z/OS uses the IODF that you were running with. If you are IPLing initially or there is no current IODF, z/OS loads wait state X'0B1', with reason code X'00B'.	When you activate a different IODF, you do not have to change LOADxx or rename IODF suffixes.
You can specify both operating system configuration ID and IODF suffix.	z/OS uses the IODF with the specified suffix for the IPL. The IODF may contain multiple operating system configurations, z/OS uses the specified one for the IPL. If an IODF with the specified suffix does not exist or does not contain the specified operating system configuration ID, z/OS enters a wait state.	When you activate a different IODF or operating system configuration, you must: • Change the suffix and configuration ID in the LOADxx member • Name the new IODF with the specified suffix and the operating system configuration with the specified ID.
You can specify an operating system configuration ID and leave the IODF suffix blank.	z/OS searches the IODFs in numerically ascending order starting with IODF00 and continuing through IODFFF until z/OS finds an IODF that contains the specified operating system configuration ID. If an IODF with the specified operating system configuration does not exist, z/OS enters a wait state.	You need to make sure that the IODF that you want z/OS to find is the first one in the sequence that has that operating system configuration ID. If you activate a different IODF, you might have to change IODF suffixes so that IODF is the first one found.

Table 6. Specifying an IODF for IPL (continued)

What You Can Specify in the IODF Statement	How z/OS Finds the IODF at IPL	How to make sure z/OS finds a new IODF at IPL
You can specify an IODF suffix and leave the operating system configuration ID blank	z/OS uses the IODF with the specified suffix as long as that IODF only contains a single OS configuration. If that IODF contains more than one operating system configuration, z/OS enters a wait state.	When you activate a different IODF, you must do one of the following: • Change the suffix in the LOADxx member • Name the new IODF with that suffix.
You can leave both the operating system configuration ID and IODF suffix blank	z/OS uses the IODF that it finds by searching in numerically ascending order starting with IODF00 for the first IODF that contains a single operating system configuration ID.	If you activate a different IODF, you do not have to change the LOADxx member but you do have to make sure that the new IODF will be the first one that z/OS finds by renaming the IODFs.
Note: If you specify asterisks, pluses, minuses, or equals, for the IODF suffix, do not change the default values for the descriptor fields when you create a production IODF. z/OS uses the default information to find the IODF.		

If you do not specify an '**', '++', '−−', or '==' for the IODF suffix, when z/OS finds the IODF, it checks whether the IODF contains a hardware I/O configuration definition that matches the representation used by the channel subsystem. Whether it matches determines the kind of configuration changes you can make. If it does match, you can change both hardware and software definitions. If it does not match, you can make only software configuration changes. To enable changes to the hardware configuration definition, you can make a software-only dynamic change that activates an IODF containing the correct hardware I/O configuration definition.

Specifying an operating system configuration ID: If you also specify an operating system configuration ID in the IODF statement, the IODF may contain multiple operating system configurations and z/OS uses the specified one. If you do not specify an operating system configuration, then the IODF should contain only one operating system configuration. If the IODF contains more than one operating system configuration, z/OS enters a wait state.

Specifying the EDT identifier: If you are letting z/OS find the correct IODF automatically and you do not specify an EDT identifier, z/OS uses '00' as the identifier. To ensure that an IODF with the identified EDT exists, you can either specify an EDT identifier and put that EDT in each IODF or put an EDT with a '00' identifier in each IODF.

Using your initial I/O configuration with special secondary (3390D) devices for reIPL

If special secondary (3390D) devices are used in subchannel set 1, the operating system must determine whether the IPL has occurred after a HyperSwap® that might have switched devices from the PPRC primary devices to the PPRC secondary devices. If so, the IPL can still use the initial I/O configuration definition, but must initialize using the correct set of devices as primaries and secondaries.

In the LOADxx parmlib member, the SCHSET parameter allows you to specify whether the special secondary (3390D) devices in subchannel set 1 are to be used as the primary devices for the IPL or the initial primary devices in subchannel set

0 are to be used. When SCHSET is set to "1," the initial PPRC special secondary devices are brought online as the new PPRC primary devices and the initial PPRC primary devices will remain offline as the new PPRC special secondary devices.

For information on specifying the SCHSET parameter of LOADxx, see *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

Defining your I/O configuration for reIPL

Defining a configuration for reIPL is an exception or recovery task. The only times you would not perform a dynamic change are:

- When you are installing a device that does not support dynamic capability
- When you are installing a new product and no device code for it is resident
- When you need to reIPL the system after a system outage

Take the following general actions to define your configuration at reIPL:

1. Reconsider the SQA and CMB parameter values in case they need to be increased due to additional devices being added to the configuration.
2. Update or create the LOADxx member so the new production IODF will be selected at IPL. This step is optional if you let the system find it by not specifying the IODF suffix. If you specify '**', '++', '−−', or '==' for the IODF suffix, z/OS uses the IODF that was the last active IODF. If you use asterisks for the IODF suffix, make sure that you always write a new IOCDS after a dynamic change and that you never change the default values of the descriptor fields when you create a production IODF.

Chapter 4. Dynamically changing an I/O configuration

HCD allows you to change an I/O configuration dynamically, that is, without an IPL or POR.

Before attempting to change an I/O configuration dynamically, ensure that your system has migrated to dynamic capability. Until your installation updates its affected application programs or installs updated products to support dynamic configuration, you should define any devices that support dynamic configuration as installation-static, so that the programs that do not support dynamic configuration can continue to use the devices.

For migration information, see “Migrating to the dynamic capability provided with HCD” on page 113.

Actions to change your configuration dynamically

Defining a configuration dynamically involves the following tasks:

1. Verify that each I/O device that you plan to delete is not needed by any application or logical partition by varying the I/O device offline.
2. Check that the current software I/O configuration definition matches the hardware I/O configuration.

   You can check that the definitions match by testing an activation of a hardware and software change. When the activation specifies configurations that do not match, the test function displays an error message. For more information about testing an activation, see “Testing dynamic activation” on page 102.
3. Attempt to change the (current) active production IODF.

   This action will cause HCD to create a new work IODF.
4. Change the work IODF to define the new I/O configuration.

   Along with specifying other information about the configuration, you use HCD to define as DYNAMIC “Yes” each device that you want to be able to dynamically change. DYNAMIC “Yes” is not required to add a device dynamically but is required to change or delete it. For more information about defining devices dynamically, “Making dynamic changes to I/O devices” on page 88.
5. Create a production IODF from the work IODF.
6. If desired, physically remove the devices that you are deleting.
7. Update or create the LOADxx member so the new production IODF will be selected at IPL. This step is optional if you let the system find it by not specifying the IODF suffix. If you specify '**', '++', '– –', or '==' for the IODF suffix, z/OS uses the IODF that was the last active IODF. If you use asterisks, ensure that you always write a new IOCDS after a dynamic change and that you never change the default values of the descriptor fields when you create a production IODF. For more information, see “IODF processing at IPL” on page 76.
8. Optionally, test the configuration before you actually activate it. For more information about testing an activation, see “Testing dynamic activation” on page 102
9. Activate the new IODF.

See “Dynamically changing an I/O configuration in LPAR mode” for a description of activating changes in logical partitions. For a software and hardware dynamic activation, the software and hardware configurations must match. When they match, activation occurs. When they do not match, activation does not occur.

See “Dynamically changing an I/O configuration in a sysplex” on page 87 for planning considerations before you dynamically activate configurations in a sysplex.

10. If desired, install the new I/O components.
11. Use HCD to write a new IOCDS for the new default IOCDS at the next POR.
12. Optionally, use HCD to create a backup copy of the new production IODF in case you lose your DASD.
13. Update the standard configuration list (CONFIGxx) member in SYS1.PARMLIB to reflect the new configuration definition.

    You can use the configuration defined in CONFIGxx to compare the differences between the current configuration and the standard configuration as defined in a CONFIGxx member. When the operator issues the DISPLAY command with the M=CONFIG(xx) option, the system displays any differences.

    Enter the DISPLAY command (D M=CONFIG(xx)) to view any differences caused by the dynamic configuration change and update the CONFIGxx member to reflect the new definition. For descriptions of the statements and parameters of CONFIGxx, see *z/OS MVS Initialization and Tuning Reference*.
14. You can also use HCD to build the CONFIGxx member from a configuration in the IODF, and to process the DISPLAY M=CONFIG(xx) command. For more information, see *z/OS HCD User's Guide*.
15. Use HCD to create a stand-alone IOCP input tape for ES/9000 PCE recovery support.

    The tape can be used for recovery if you lose the disk holding the IOCDS in the processor controller. For a description of creating an IOCP input data set, see *z/OS HCD User's Guide*.

Related information:
- For more information for the type of I/O component you are defining, see “Dynamically changing component definitions” on page 88.
- For the recovery steps to take if your system fails during a dynamic configuration change, see “Recovery during dynamic configuration” on page 103.

Dynamically changing an I/O configuration in LPAR mode

When a processor is running in LPAR mode, any single logical partition can make dynamic configuration changes that affect the hardware configuration for all the other logical partitions. To eliminate unwanted disruption across logical partitions (to not delete hardware I/O components that are in use by another logical partition), you must follow these steps:

1. In all logical partitions ***except one***, use HCD panels to make software-only changes. (See “Making software-only dynamic changes” on page 8.) Think of the software only changes in these partitions as step 1. In the HCD panels, activate the “software definitions only and validate hardware changes” option.

You can also use the ACTIVATE command with the SOFT option. See *z/OS MVS System Commands*. As you make the software-only changes to each logical partition, the system validates that the I/O components being deleted are not in use.

If you are certain that a dynamic configuration change will not affect a particular logical partition, you do not need to make software-only changes to that logical partition.

2. In the ***remaining*** one logical partition, use HCD panels to make both hardware and software changes. (See "Making both hardware and software dynamic changes" on page 8.) Think of the hardware and software change in this remaining partition as step 2. In the HCD panels, activate "hardware and software definitions". You can also use the ACTIVATE command and omit the SOFT option.

Dynamic I/O configuration considerations

Consider the following when making dynamic I/O configuration changes:

- **Ensuring that you activate the correct IODF in Step 1:** When you make dynamic changes to coupling facility devices and control units be careful to activate the correct IODF in step 1. If you activate an incorrect IODF and the system displays message IOS500I or IOS1500I with a note indicating that the IODF does not contain a configuration definition that matches the current hardware definition, you must perform the following steps:
  1. In the step 1 partition, activate the previous IODF to reset the configuration definition to its previous state.
  2. In the step 1 partition, activate the correct IODF.
  3. In the step 2 partition, perform the software and hardware change.

  If the system does not display the note on the IOS500I or IOS1500I message, you can activate the correct IODF in the step 1 partition and perform the software and hardware change in the step 2 partition.
- **Dynamic deletion of I/O hardware components requires FORCE:** To prevent you from inadvertently deleting I/O components that another logical partition is using, HCD only allows you to delete I/O components when you specify "Yes" in the "Allow hardware deletes (FORCE)" field on the Activate New Configuration panel. An activation that deletes or modifies I/O components may result in deletion of a resource - in this case, you must use the FORCE parameter on the activation. See *z/OS MVS System Commands*.
- **Dynamic deletion of I/O and coupling facility hardware components:** If you are deleting I/O and coupling facility hardware components that another logical partition is using, HCD only allows you to delete I/O components when you specify "Yes" in the "Allow hardware deletes (FORCE)" field on the Activate New Configuration panel. You can also use the ACTIVATE command to either dynamically test or activate a new I/O configuration definition. To delete components, use the FORCE attribute on the ACTIVATE command. See *z/OS MVS System Commands*.
- **Dynamic activation in a sysplex that includes a uniprocessor:** If you use the ACTIVATE command in a sysplex that includes a uniprocessor, that system might encounter problems in maintaining XCF synchronization with the rest of the sysplex.

Example of deleting an I/O device in LPAR mode

For the sample processor running in LPAR mode shown in Figure 23 on page 86, assume the following:

- You wish to delete device 000F.
- The current production IODF is SYS1.IODF14.

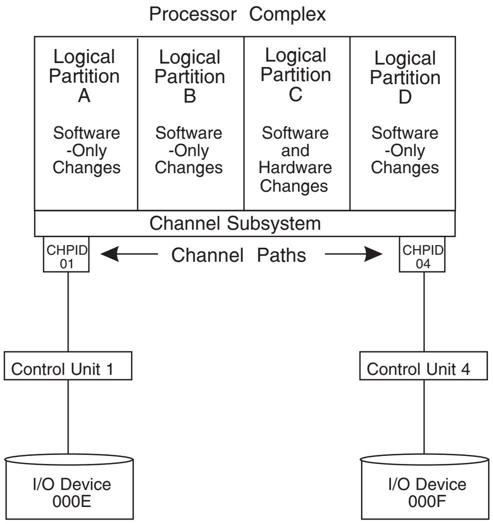


Figure 23. Sample processor running in LPAR mode

In this example, you would make dynamic changes through the following steps:

1. Ensure that device 000F supports dynamic configuration.
2. Ensure that device 000F has been defined as dynamic.
3. Use HCD panels to create a new production IODF (SYS1.IODF15) that omits the device F definition. For more information, see *z/OS HCD User's Guide*.
4. Make software-only dynamic changes in logical partition A. (See “Making software-only dynamic changes” on page 8.)
5. Make software-only dynamic changes in logical partition B.
6. Make software-only dynamic changes in logical partition D.
7. Make hardware and software dynamic changes in logical partition C. (See “Making both hardware and software dynamic changes” on page 8.)
8. Physically remove device F.

Dynamically adding and removing logical partitions to the I/O configuration

On a z990 processor with a supporting level of microcode or on a z890 processor, you can pre-define one or more reserved logical partitions, which you can later add and remove from the I/O configuration dynamically.

- To add a logical partition dynamically to the I/O configuration, you first pre-define it, which reserves the space in the IOCDS and HSA. Then later, you can assign a name and resources to the logical partition using dynamic I/O.
- To delete a logical partition dynamically from the I/O configuration, you unassign the resources to it and then unassign the partition name. The partition will then revert back to the pool of reserved logical partitions - the space used by the partition is not returned to the available pool of storage.

Consider the following when dynamically adding and removing logical partitions to the I/O configuration:

- You cannot dynamically remove an active logical partition from the I/O configuration.
- After you assign a name to a partition, you can assign dedicated CHPIDs for that partition and add the partition to the candidate lists of any reconfigurable, non-managed shared, or spanned CHPIDs. Managed channels do not have to be added to the candidate list - they are in the candidate list, even unnamed ones, by default.
- Before deleting a logical partition from the I/O configuration by unassigning the partition name, you must unassign any dedicated CHPIDs defined for the partition . If there are any reconfigurable, non-managed shared, or spanned CHPIDs defined for the partition, you must remove the partition name from the candidate list for those CHPIDs.
- When you prepare to delete a logical partition dynamically, if Cryptographic coprocessors are configured to the logical partition you are trying to delete, these features must be deconfigured from the LPAR image using the zSeries Hardware Management Console or Support Element tasks. For more information about how to view or change your Cryptographic assignments, see the following documentation for your installed hardware:
  - *Support Element Operations Guide*
  - *Processor Resource/Systems Manager™ Planning Guide*
- When you delete a logical partition from the I/O configuration dynamically, you do not have to delete the partition from each device's candidate list , unless you used an explicit candidate list when you defined the device.

Dynamically changing an I/O configuration in a sysplex

You can use HCD as a single point of control to dynamically activate an I/O configuration on one or more systems configured in a sysplex. To determine the supported processors, run the HCD Supported Hardware Report. For information about running reports, see *z/OS HCD User's Guide*.

Using the “Activate configuration sysplex-wide” option on the HCD Activate and Process Configuration Data panel, you can make software-only changes as well as software and hardware changes to one or more systems from a system in your sysplex.

Consider the following requirements before you activate your sysplex-wide configuration changes:

- You can perform a sysplex-wide activation only on a system that is running and is configured in the sysplex couple data set. For more information about sysplex couple data sets, see *z/OS MVS Setting Up a Sysplex*.
- For a sysplex-wide activation to work on a system, the system must allow software configuration changes. To verify that a system allows software configuration changes, select the “Activate or process configuration data” option on the HCD Primary Task Selection panel and then the “View active configuration” option on the HCD Activate and Process Configuration Data panel on that system. HCD displays the currently active IODF, the HSA token, and information about limitations to the activation scope.

After you initiate a sysplex-wide activation, respond to any activation messages.

Dynamically changing component definitions

You can dynamically add, modify, or delete components in the configuration. You can make software and hardware changes or software-only changes. In most cases, you will make simultaneous dynamic configuration changes to both hardware and software configuration definitions. The following sections describe how to dynamically change component definitions. For software-only changes, you do not have to perform all the steps. It is noted when the step is not required for a software-only change.

Any dynamic configuration addition, modification, or deletion requires the following two general steps:

- Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. See *z/OS HCD User's Guide*.
- Activate the new IODF.

  You can use either HCD panels or the ACTIVATE command to dynamically activate or test a new I/O configuration definition. Generally, system programmers use HCD and system operators use the ACTIVATE command. To use HCD panels, see *z/OS HCD User's Guide*. To use the ACTIVATE command, see *z/OS MVS System Commands*.

  Notes:

  1. For software/hardware changes, activate “hardware and software definitions”. Also, write a new I/O configuration data set (IOCDS) for the new default IOCDS at the next POR.
  2. For software-only changes, activate “software definitions only.” You can also choose to “validate hardware changes,” which validates that I/O components being deleted are not in use.

Additional steps related to specific I/O components are described in

- “Making dynamic changes to I/O devices”
- “Making dynamic changes to channel paths” on page 92
- “Making dynamic changes to control units” on page 95

Coupling facility device and control unit definitions in the IODF are created or deleted automatically when coupling facility channel paths are connected or disconnected. See “Making changes to coupling facility devices and control units” on page 97.

Making dynamic changes to I/O devices

You can dynamically add, modify, or delete device definitions. Remember that devices defined as installation-static can still be dynamically added.

Dynamically adding I/O devices

To dynamically add an I/O device, use the following general steps:

1. For each device that you plan to add, ensure that its device support code was loaded at the last IPL.

   There are two ways which device support code is loaded at IPL. The choice made at the last IPL can determine whether the device support code is loaded for the device that you want to add. Through the IODF statement in the LOADxx member of SYS1.PARMLIB, or SYSn.IPLPARM, for the last IPL, you might have chosen either of the following:

   - To load, at IPL, all the dynamic device support code in SYS1.NUCLEUS.

This choice, which is the default, allows you to dynamically add any device that supports dynamic configuration.

- To load, at IPL, device support code for only those device types defined in the IODF. This choice allows you to dynamically add only those device types that were defined in the IODF.

  If you add a new device and the device support code for that new device was not loaded at IPL, your system cannot use that device. Before your installation can use that device, you must reIPL the system to load the device support code.

After a HyperSwap failover swap has occurred, ensure that if a Special Secondary (3390D) device is being added, a 3390B PPRC primary device with the same four-digit device number is also being added. If one device in a Special PPRC pair (3390B/3390D) is not being added when the other device in the pair is being added, the dynamic activate will fail.

For details about specifying your choice at IPL, see "IODF processing at IPL" on page 76.

2. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Including an I/O component in an IODF adds that I/O component's definition to the I/O configuration definition.
3. Activate the new IODF.
4. Install the new I/O devices that you have dynamically added through the new IODF. As required, format the I/O devices.

Besides these general actions to add a device, other actions might be required to make certain devices usable. For example, adding a direct access storage device (DASD) to support a page data set requires the following actions:

1. For the DASD that you plan to add, ensure that device support code was loaded at the last IPL.
2. Create a new I/O definition file (IODF) that defines the DASD to both hardware and software.
3. Activate the new IODF.
4. Install the DASD.
5. Run the ICKDSF program to format a volume on the DASD.
6. Vary the DASD online to z/OS.
7. Run IDCAMS to define the page data set on the DASD.
8. Enter the PAGEADD command to add the page data set.

z/OS directs paging activity to the volume.

Note: You can add a new bank of DASD using the same procedure, but perform the steps for all the DASDs in the bank.

Dynamically deleting I/O devices

To dynamically delete an I/O device, use the following general steps:

1. Ensure that each device you plan to delete is not needed by any application running in the processor. In a Processor Resource / System Management (PR/SM) environment, this includes applications running in any logical partition in your processor. See "Dynamically changing an I/O configuration in LPAR mode" on page 84.
2. Ensure that each device you plan to delete is offline and not in use. CTC and TP devices must also not be allocated. For JES3 FSA (Functional Subsystem Application) devices, HCD will allow the deletion as long as the device you are

deleting is offline. However, JES3 requires that all FSA devices assigned to the FSS (Functional SubSystem) are inactive when performing a subsequent Hot Start with Refresh to inform JES3 that FSA devices have been deleted from the IODF configuration.

After a HyperSwap failover swap has occurred, ensure that if a Special Secondary (3390D) device is being deleted, a 3390B PPRC primary device with the same four-digit device number is also being deleted. If one device in a Special PPRC pair (3390B/3390D) is not being deleted when the other device in the pair is being deleted, the dynamic activate will fail.

3. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Omitting an I/O component from an IODF deletes that I/O component's definition from the I/O configuration definition.
4. Physically remove the I/O devices that you have dynamically deleted through the new IODF.

   Note: This step is not required for a software-only change.
5. Activate the new IODF.

Besides these general actions to delete a device, other actions might be required for certain devices. For example, deleting a DASD that contains only one page data set requires the following actions:

1. Ensure that the DASD you plan to delete is not needed by any application.
2. Create a new IODF that ***does not*** define the DASD.
3. Enter the PAGEDEL command.

   The system displays message IEE206I to indicate that the page delete request is accepted and in progress.

   When the delete request completes, the system displays message IEE205I to indicate that the page data set is deleted.
4. Enter the DISPLAY UNITS command to determine whether the device that held the page data set is unallocated.
5. After the device is unallocated, vary it offline to z/OS.
6. Physically remove the DASD.
7. Activate the new IODF.

This procedure deletes the DASD definition from z/OS and HSA.

Note: You can delete a bank of DASD using the same procedure, but perform the steps for all the DASDs in the bank.

Dynamically changing I/O devices

To dynamically change a device, use the following general steps:

1. For devices that you plan to change, if you are changing certain fields in the IODF, you must vary devices offline.

   To guard against losing I/O access unexpectedly, you must vary offline I/O devices before you activate an IODF that changes the characteristic fields described in Table 7 on page 91. You might need to cancel applications that have a device allocated to get the device offline and TP and CTC devices must also be unallocated. To change the number of alias UCBs, you must change the control unit type and the device type of all the devices attached to the control unit at the same time. This action requires that all the devices for the control unit are offline at the same time.

Table 7. Dynamic I/O Offline Requirements for Device Characteristic Fields

Vary the affected I/O components offline before activating a new IODF that changes these device characteristics:	**Before activating the new IODF, take the following actions:**
• Unit Address • Device Type	In each logical partition that can access the I/O device, vary the I/O device offline.
• Connected to CUs	In each logical partition that will lose access to this I/O device, vary the I/O device offline. In each logical partition that will be able to access the I/O device through other control units, vary offline only the logical paths associated with the control unit.
• Device Candidate List	In each logical partition that will lose access to this I/O device, vary the I/O device offline.

You do **not**, however, have to vary an I/O device offline before activating an IODF when you change any of the following I/O device characteristics:
- Serial Number
- Description
- TIME OUT
- STADET
- Preferred CHPID

2. After a HyperSwap failover swap has occurred, if a Special Secondary (3390D) device is being changed, ensure that a 3390B PPRC primary device with the same four-digit device number is also being changed. If one device in a Special PPRC pair (3390B/3390D) is not being changed when the other device in the pair is being changed, the dynamic activate will fail.
3. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD.
4. Activate the new IODF.

Limitations on dynamic changes to I/O devices

Even though a device supports dynamic configuration, the system environment might not allow dynamic changes that would add, modify, or delete the device definition.

- You cannot dynamically add certain devices. These include devices for which device support code was not loaded at IPL.
- You cannot dynamically modify or delete certain devices. These include devices containing data sets that you must never unallocate, such as the device containing the system residence volume or a device that contains a data set in the link list concatenation. Examples of data sets that you must never unallocate include:
  - Common page data sets
  - LPA page data sets
  - SYS1.LINKLIB
  - SYS1.MIGLIB
  - The logrec data set
  - The VIO journaling data set

  Other factors that determine whether or not you can use dynamic configuration for a device include:
  - A product or program can use a device without accessing it through allocation. If a device supports dynamic configuration, but a product or

program that uses that device does not support dynamic configuration, you must not define that device as dynamic. See “UCB services considerations” on page 113.

Note: Do not define devices that contain control volumes (CVOLs) as dynamic devices; define them only as installation-static.

– If you plan to use a pair of devices in a dynamic device reconfiguration (DDR) swap, both devices must have the following two attributes in common:
  - Whether the device type supports dynamic configuration
  - Whether the device is defined as dynamic

Any dynamic change that causes a device's UCB to be deleted and added again, causes the device's MIH time interval to be reset to the default MIH setting for its device class. To reestablish the previous MIH interval, enter the SETIOS MIH command when the dynamic change is completed. For information about the SETIOS MIH command, see *z/OS MVS System Commands*.

Console devices: You can define console devices as dynamic; however:

- You can dynamically delete or modify any device that was defined as a console in the CONSOLxx member of SYS1.PARMLIB at IPL only after the console definition has been removed from the system by IEAVG730 or IEARELCN. For information about removing a console through these methods, see *z/OS MVS Planning: Operations* .
- You cannot use any dynamically added device as an MCS console.

Considerations for z/OS installed products

Certain products installed with z/OS require that you take additional steps to define a configuration change to the product. If you do not take these additional steps before making a dynamic configuration change, the configuration defined to these products might be out of synchronization with the z/OS configuration definition.

After you dynamically add an I/O device to z/OS, define it to other products through their specific interfaces for defining I/O configurations. For example, to add a Virtual Telecommunications Access Method (VTAM) device:

1. Dynamically add the VTAM device to z/OS.
2. Create a new telecommunication program (TP) topology that includes the VTAM device you are adding.
3. Define that new TP topology to VTAM.

When you delete an I/O device definition, first remove the definition from the related product, then delete the I/O device from z/OS by activating a new configuration definition. For example, to delete a VTAM device:

1. Create a new TP topology that does not include the device you are deleting.
2. Define that new TP topology to VTAM.
3. Dynamically delete the VTAM device from z/OS.

Making dynamic changes to channel paths

You can add, modify, and delete channel path definitions. If you define a device as static or installation-static, you cannot delete or modify the channel path definition to that device.

Using an IBM zSeries 900 or later processor, you will be able to dynamically add, delete, or modify coupling facility receiver channels and coupling facility peer channels. For an S/390 9672 Parallel Transaction Server, you cannot dynamically add, delete, or modify coupling facility receiver (CF receiver) channels. However, at POR, you could define more CF receiver channels than are currently connected to coupling facility sender channels. Later, to create more connections from an operating system to the coupling facility, you can activate an IODF that dynamically adds CF sender channels connected to the defined CF receiver channels.

Dynamically adding channel paths

To dynamically add a channel path, use the following general steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Including a channel path in an IODF adds that I/O component's definition to the I/O configuration definition.
2. Activate the new IODF.
3. Install the new I/O components that you have dynamically added through the new IODF. As required, configure the channel paths.

Dynamically deleting channel paths

How you dynamically delete a channel path depends on whether or not that channel path shares any control units and devices with one or more other channel paths.

To delete a channel path that is the only channel path attached to all of its devices and control units, use the following steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Create an IODF that defines neither the channel path nor its devices and control units.
2. Ensure that the channel path and its devices and control units are not in use by any application.
3. Vary all I/O devices on the channel path offline to z/OS.
4. Configure the channel path offline to z/OS.
5. Physically remove the channel path and all I/O devices and control units on the channel path.
6. Activate the new IODF.

To delete a channel path that is ***not*** the only channel path attached to its devices and control units, use the following general steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Create an IODF that meets both of the following requirements:
   - The new IODF does not define the channel path or the devices and control units attached exclusively on the channel path.
   - The new IODF does define those device and control units also attached on other channel paths.
2. Ensure that no devices and control units on this channel path are in use by this channel path.
3. For all devices not exclusively on this channel path, vary offline to z/OS all device paths through this channel path.
4. Vary offline to z/OS all devices exclusively on this channel path.
5. Configure the channel path offline to z/OS.

6. Physically remove the channel path and the I/O devices and control units ***exclusively*** on this channel path.
7. Activate the new IODF.

There is a consideration when you delete CF sender channel paths. If a coupling facility is defined in the coupling facility resource management (CFRM) policy and has working connections from a z/OS system to the coupling facility, you cannot make dynamic changes that cause a loss of connectivity to the facility. For example, you cannot dynamically delete the last operational coupling facility channel path.

To delete the last channel path connected to the coupling facility, use the following general steps:

1. If the coupling facility is in the current CFRM policy, activate a new policy that does not specify the coupling facility. For information about setting up and activating the CFRM policy, see *z/OS MVS Setting Up a Sysplex*.
2. Configure the channel path offline to z/OS.
3. Physically remove the channel path.
4. Activate a new IODF.

Dynamically changing channel paths

To dynamically change a channel path, use the following general steps:

1. For channel paths that you plan to change, if you are changing **certain** fields in the IODF, you must configure channel paths offline.

   To guard against losing I/O access unexpectedly, you must vary offline I/O devices before you activate an IODF that changes the characteristic fields described in Table 8:

Table 8. Dynamic I/O Offline Requirements for Channel Path Fields

Vary the affected components offline before activating a new IODF that changes these channel path characteristics:	**Before activating the new IODF, take the following actions:**
• Channel Path Type • Dynamic Switch ID • Operation Mode	In all logical partitions in your processor complex, configure the channel path offline. If the channel path is the last path to an I/O device, you must vary the device offline before you can configure the channel path offline.
• Channel Path Access List • Channel Path Candidate List	In each logical partition that will lose access to I/O devices on this channel path, do the following: 1. For I/O devices that will become inaccessible to the logical partition, vary those I/O devices offline. 2. For I/O devices that will remain accessible by the logical partition through other channel paths, vary offline only the logical paths associated with this channel path. 3. Configure the channel path offline. **Note:** For coupling facility sender channel paths, you must first vary the CF sender channel path offline to the coupling facility. In all logical partitions that will gain access or continue accessing this channel path, you need not configure or vary anything offline.

| **Note:** This table is also valid for spanned channels.

You do **not**, however, have to vary an I/O device offline before activating an IODF when you change any of the following I/O device characteristics:
- Entry Switch Number
- Entry Port
- Description

2. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD.
3. Activate the new IODF.

Note: You usually cannot remove a logical partition from a channel path candidate list unless the channel path is offline. You can, however, remove a logical partition, other than your own, from the channel path candidate list for a reconfigurable or shared ESCON connection (CNC and CTC) or coupling facility sender channel path while the channel path is online to the other logical partition if you perform one of the following actions:

- Use the ACTIVATE command with the FORCE(CANDIDATE) parameter.
- On the HCD Activate New Hardware and Software Configuration panel, specify “Yes” in the “Delete partition access to CHPIDs unconditionally (FORCE=CANDIDATE)” field.

Making dynamic changes to control units

You can add, modify, and delete control unit definitions.

Dynamically adding control units

To dynamically add a control unit, use the following general steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Including an I/O component in an IODF adds that I/O component's definition to the I/O configuration definition.
2. Activate the new IODF.
3. Install the new I/O components that you have dynamically added through the new IODF. As required, configure the channel paths, format the I/O devices, and vary the I/O components online.

Dynamically deleting control units

z/OS verifies that the control units are not in use (that the set of all device paths are offline) before deleting the control unit.

How you dynamically delete a control unit depends on whether or not that control unit shares any devices with one or more other control units.

To delete a control unit that is the only control unit attached to ***all*** its devices, use the following general steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Create an IODF that defines neither the control unit nor its attached devices.
2. Ensure that none of the devices on the control unit are in use by any application.
3. Vary all devices on the control unit offline to z/OS.
4. Physically remove the control unit and all its attached devices.
5. Activate the new IODF.

To delete a control unit attached to any devices that are ***also*** attached to one or more other control units, use the following general steps:

1. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD. Create an IODF that meets both of the following requirements:
   - The new IODF does not define the control unit or its exclusively attached devices.
   - The new IODF does define those devices also attached to other control units.
2. Ensure that none of the devices exclusively on this control unit are in use by any application.
3. Vary offline to z/OS all devices exclusively on the control unit. For all devices also attached to other control units, vary offline to z/OS all device paths through this control unit.
4. Physically remove the control unit and all its exclusively attached devices.
5. Activate the new IODF.

Dynamically changing control units

To dynamically change a control unit, use the following general steps:

1. For control units that you plan to change, if you are changing particular fields in the IODF, you must vary attached devices offline.

   To guard against losing I/O access unexpectedly, you must vary offline I/O devices before you activate an IODF that changes the characteristic fields described in Table 9:

   Table 9. Dynamic I/O Offline Requirements for Control Unit Fields

Vary the affected I/O components offline before activating a new IODF that changes these control unit characteristics:	**Before activating the new IODF, take the following actions:**
• Control unit attachment type • Logical Address • Protocol • I/O Concurrency Level	• Vary offline all I/O devices connected to the control unit. This is required when the attachment type changes, for example, from ESCON to FICON or from serial to parallel.
• Channel Path IDs • Link Address	In each logical partition that will lose access to I/O devices on this control unit, do the following: 1. For I/O devices that will become inaccessible to the logical partition, vary those I/O devices offline. 2. For I/O devices that will remain accessible by the logical partition through other channel paths, vary offline only the logical paths that you are removing from this control unit.
• Unit Addresses (with Ranges and Numbers of Units)	Before deleting a unit address, vary its associated I/O device offline.

   You do not, however, have to vary an I/O device offline before activating an IODF when you change any of the following control unit characteristics:
   - Serial Number
   - Description
2. Describe the new I/O configuration definition in an I/O definition file (IODF) that you create through HCD.
3. Activate the new IODF.

Making changes to coupling facility devices and control units

Coupling facility devices and control units are defined to represent the connections in the channel subsystem connection to a coupling facility. The coupling facility control unit and device definitions are created in an IODF when you connect CF sender and CF receiver channel paths or two CF peer channel paths using HCD. One coupling facility control unit and two coupling facility devices are defined when you first connect a CF sender channel path to a CF receiver channel path for a coupling facility. Each subsequent channel connection adds two more coupling facility device definitions. For a CF peer channel connection, one control unit and seven CF device definitions are generated on one or on both sides of the connection depending on whether the sending function is used on only one or on both sides of the connection. You do not explicitly add or delete coupling facility devices and control units in the IODF; they are added and deleted based on the CF sender-to-receiver or peer-to-peer channel connections that you make.

To use the coupling facility after you make the connections, a CFRM policy containing the coupling facility must be active. For information about setting up and activating a CFRM policy, see *z/OS MVS Setting Up a Sysplex*.

A CF sender-to-CF receiver or peer-to-peer connection can cause changes to a coupling facility control unit definition. The first path you connect adds the coupling facility control unit and the last path you disconnect deletes the coupling facility control unit, but all other connections and disconnections of channel paths change the control unit definition.

The following sections describe how dynamic changes are made to coupling facility devices and control units when you connect and disconnect channel paths for a processor.

Making the first channel path connection for a coupling facility

When you connect the first CF sender channel path to a CF receiver channel path for a coupling facility, HCD creates a coupling facility control unit and two coupling facility device definitions in an IODF. When you connect the first two CF peer channel paths for a coupling facility, HCD creates a coupling facility control unit and seven coupling facility device definitions in an IODF.

Perform the following steps when you connect the first CF sender channel path or CF peer channel path:

1. In HCD, connect a CF sender channel path to a CF receiver channel path, or connect two CF peer channel paths.
2. Accept or change the control unit and device numbers HCD proposes.
3. Activate the new IODF.
4. Install the channel path.
5. To allow the system to use the coupling facility, activate a coupling facility resource management (CFRM) policy that defines the coupling facility, if one is not already activated.
6. Configure the channel path online.

Making subsequent channel path connections for a coupling facility

When you make subsequent CF sender-to-CF receiver channel path connections for a coupling facility, HCD creates two coupling facility device definitions in the IODF. When you make subsequent CF peer-to-peer channel path connections for a coupling facility, HCD creates seven coupling facility device definitions in the

IODF. For both the CF sender channel path connection and the CF peer channel path connection, HCD also changes the existing coupling facility control unit definition to reflect the new channel path connection in the IODF.

Perform the following steps when you connect subsequent CF sender or CF peer channel paths:

1. In HCD, connect a CF sender channel path to a CF receiver channel path, or connect two CF peer channel paths.
2. Accept or change the device numbers proposed by HCD.
3. Activate the new IODF.
4. Install the channel path.
5. Configure the channel path online.

Breaking a channel path connection that is not the last connection to a coupling facility

When you disconnect a CF sender channel path connection that is not the last connection for a coupling facility, HCD deletes two coupling facility device definitions in the IODF. When you disconnect a CF peer channel path connection that is not the last connection for a coupling facility, HCD deletes seven coupling facility device definitions in the IODF. For both the CF sender channel path disconnection and the CF peer channel path disconnection, HCD also changes the existing coupling facility control unit definition to reflect the channel path disconnection in the IODF.

Perform the following steps when you disconnect a CF sender or a CF peer channel path that is not the last channel path connection to the coupling facility:

1. Disconnect a CF channel path from the CF receiver channel path, or disconnect the two CF peer channel paths.
2. Configure the disconnected channel path offline.
3. Activate the new IODF.

Breaking the last channel path connection to a coupling facility

When you disconnect the last CF sender channel path connection for a coupling facility, HCD deletes a coupling facility control unit and two coupling facility device definitions in the IODF. When you disconnect the last CF peer channel path connection for a coupling facility, HCD deletes a coupling facility control unit and seven coupling facility device definitions in the IODF.

Note: If a coupling facility is defined in the active coupling facility resource management (CFRM) policy, you cannot dynamically delete the coupling facility control unit by disconnecting the last channel paths.

Perform the following steps when you disconnect the last CF sender channel path or the last CF peer channel path from a coupling facility:

1. Quiesce activity to the coupling facility by activating a CFRM policy that does not contain the coupling facility.
2. Disconnect the last CF sender or CF peer channel path from a processor to a coupling facility in the IODF definition.
3. Activate the new IODF.

Dynamic I/O configuration in a multiple LCSS environment

You can use multiple logical channel subsystems (LCSS) on an IBM zSeries 890, zSeries 990, System z9 or System z10 platform. A z/OS system at the appropriate exploitation support level can do a dynamic ACTIVATE of hardware for any LCSS. However, a system running on a z890 or z990 with compatibility support installed can only use a dynamic ACTIVATE to make hardware changes to an LCSS of zero (with some restrictions, as shown in the figures in this topic).

If you must make a hardware change to anything in LCSS other than 0, you must do a power-on reset (POR), or do the hardware ACTIVATE from a system that has the z890/z990/z9109 exploitation support installed.

Figure 24 shows a system on a z990 with compatibility support installed. This system can do a dynamic ACTIVATE to make hardware changes to LCSS 0, but not to LCSS 1:

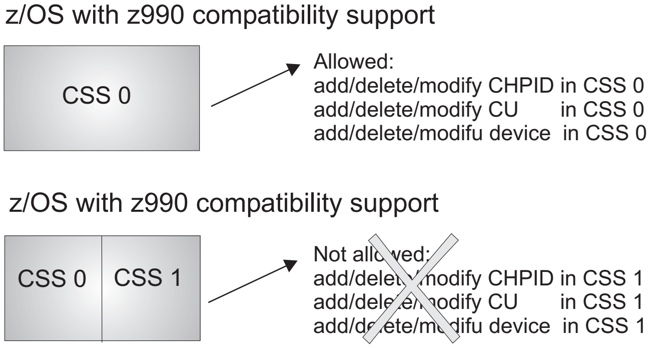
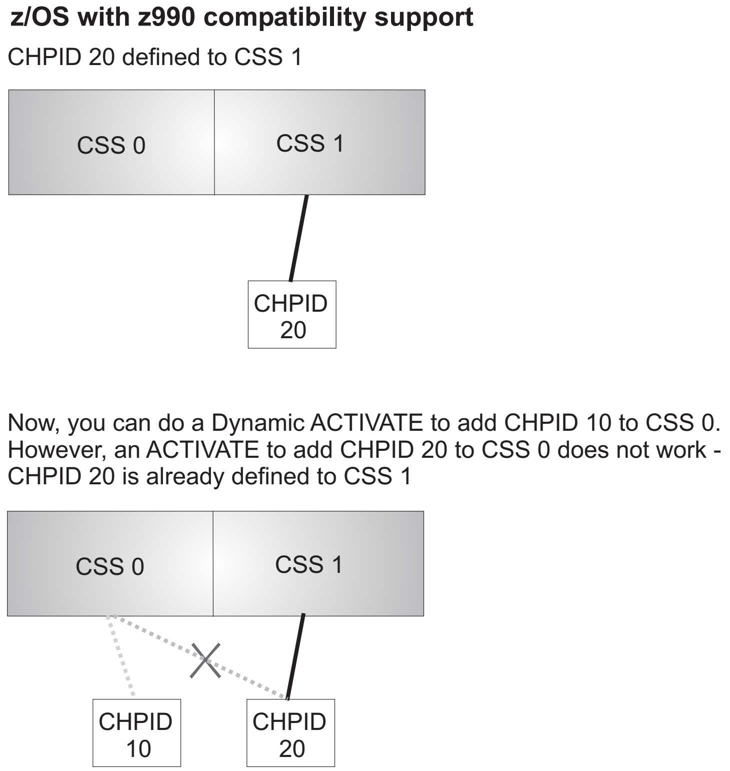

z/OS with z990 compatibility support
CSS 0
Allowed:
add/delete/modify CHPID in CSS 0
add/delete/modify CU in CSS 0
add/delete/modifu device in CSS 0
z/OS with z990 compatibility support
CSS 0
CSS 1
Not allowed:
add/delete/modify CHPID in CSS 1
add/delete/modify CU in CSS 1
add/delete/modifu device in CSS 1

Figure 24. Dynamic I/O reconfiguration management on a z990 with compatibility support

If you have defined two LCSSs, you can add resources using a dynamic ACTIVATE to LCSS 0, as long as the resources are not shared (spanned) with LCSS 1. Figure 25 on page 100 shows how you can add CHPID 10 to LCSS 0. It also shows that you **cannot** add CHPID 20 to LCSS 0, because it is shared with LCSS 1 (spanned). For more information, see “Defining spanned channel paths” on page 35.



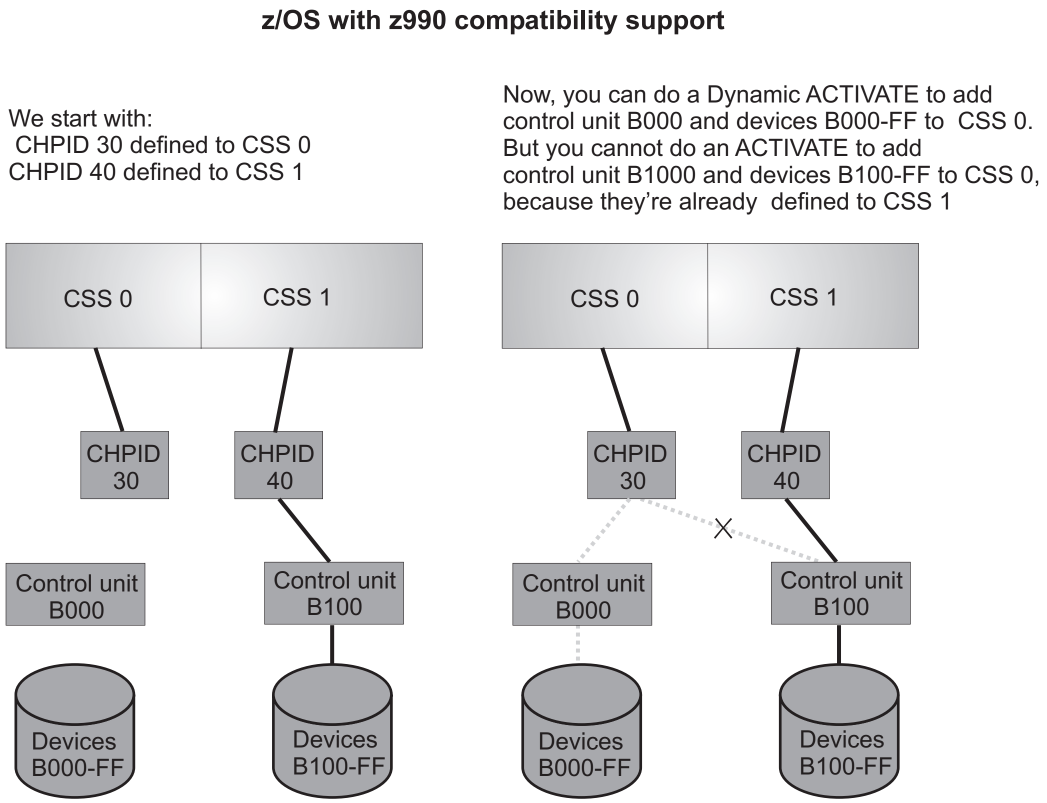
z/OS with z990 compatibility support

CHPID 20 defined to CSS 1

Now, you can do a Dynamic ACTIVATE to add CHPID 10 to CSS 0. However, an ACTIVATE to add CHPID 20 to CSS 0 does not work - CHPID 20 is already defined to CSS 1

Figure 25. Dynamically adding CHPIDs in a two-LCSS environment on a z990 with compatibility support

Figure 26 on page 101 shows how you can add control units and devices to LCSS 0, unless they are already defined to LCSS 1:



z/OS with z990 compatibility support

We start with:
CHPID 30 defined to CSS 0
CHPID 40 defined to CSS 1

Now, you can do a Dynamic ACTIVATE to add control unit B000 and devices B000-FF to CSS 0. But you cannot do an ACTIVATE to add control unit B1000 and devices B100-FF to CSS 0, because they're already defined to CSS 1

Figure 26. Dynamically adding control units and devices in a two-LCSS environment on a z990 with compatibility support

Figure 27 on page 102 shows how you can change some resource attributes for LCSS 0. You can:

- For CHPID 20, you can do an ACTIVATE to change candidate or access list for LCSS 0. You cannot remove the last partition from the candidate list.
- For control unit B000, you use an ACTIVATE to add or delete CHPIDs. You cannot remove or change all the CHPIDS for a control unit.
- For device B000–FF, you can use an ACTIVATE to change the candidate list or preferred path. You cannot remove the last partition from the candidate list for a device.
- If a control unit is already defined to LCSS 0, you can add devices to it, even if they are shared with LCSS 1.

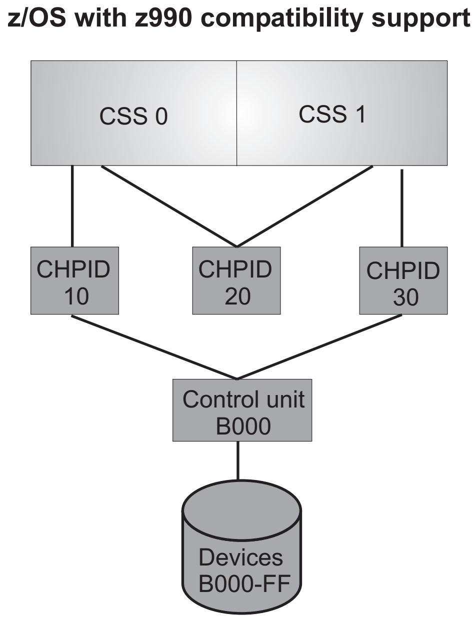


Figure 27. Dynamically adding shared resources in a two-LCSS environment on a z990 with compatibility support

Testing dynamic activation

When you dynamically activate a new I/O configuration, you can choose to first run in test mode, to see if the activation will run. If the test indicates a problem, HCD displays error messages at the terminal.

The test function includes the following checks:

- The IODF specified for activation exists and is a production IODF.
- The IODF is allocated in a single extent. Otherwise, HCD issues message CBDA009I to warn you that IPLing with a multiple extent IODF would result in a system wait state.
- The EDT specified for activation exists and is contained in the specified IODF.
- The new configuration information will fit into the current HSA.
- The device support code for all devices is present.
- The device support code for a device supports dynamic capability.
- I/O devices that you are deleting are offline.
- Paths that you are deleting are offline.
- The software activation specifies an IODF that matches the configuration definition in HSA.
- The specified IOCDS is a valid data set.
- The activation is not changing partition names or the number of partitions.
- Any new processor has at least one CHPID identified for it.
- The activation does not change a processor type or configuration mode.
- The activation does not cause a physical control unit to be added to a logical control unit if the addition causes two logical control units (with devices attached) to merge into one logical control unit.

- The activation does not cause a physical control unit to be removed from a logical control unit if the deletion causes a split of one logical control unit (with devices attached) into two logical control units.
- The activation does not change a coupling facility receiver channel path.
- The activation does not cause a loss of connectivity to an active coupling facility by deleting the last operational channel path from the coupling facility control unit.

You can run with the test function before you activate using the HCD panels or ACTIVATE command. On the HCD Activate New Configuration panels, there is a TEST Only field in which you can specify YES to test the activation and NO to when you ready to perform the activation. To use the test function with the ACTIVATE command, specify the TEST keyword. Omit the TEST keyword to actually perform the activation.

There are some checks that the test function does not perform that the actual activation does:

- The test function does not check for pinned devices because devices are only unpinned for the actual delete signal. This is more efficient. It is also possible that during a test the device would not be pinned and then attention processing causes the device to be pinned during actual activate.
- The test function also does not check SQA storage requirements for changes due to activate on the test.

If you activate hardware and software in test mode, HCD can provide information about channel paths, control units, and devices that are deleted, modified, or added. To obtain this information, you must either:

- Set the HCD profile option SHOW_IO_CHANGES=YES in the HCD dialog, or
- Enter the ACTIVATE command using the activate action in the HCD dialog.

Recovery during dynamic configuration

If your system fails while making dynamic configuration changes to the hardware definition, you can recover the I/O configuration even if the system failure requires a reIPL.

Notes:

1. Although these recovery procedures describe using the ACTIVATE command, you can also use an HCD panel to recover from a failure that occurs during dynamic configuration changes. When HCD recommends recovery during configuration activation, a panel displays where you can specify recovery options. You can choose to try to continue with the activation or to reset the configuration to the original (source) configuration. For more information, see *z/OS HCD User's Guide*.
2. If your processor is running in logically partitioned (LPAR) mode, see "Recovering LPAR mode dynamic configuration changes" on page 104.

Recovering dynamic configuration changes

After a failure during a dynamic configuration change, you can either recover the configuration that was active before the failure or continue making changes from the point of failure and complete the configuration change. In the latter case, z/OS must have access to both the source IODF, which defines the original I/O hardware configuration, and the target IODF, which defines the new I/O hardware configuration.

Recovering original I/O configuration without reIPL

If the system failure does not require a reIPL, and you want to recover the I/O configuration that was active before the failure, enter the ACTIVATE command with the RECOVER keyword and SOURCE option. z/OS backs out all dynamic I/O hardware changes made prior to the failure, returning the system to its original (source) I/O hardware configuration.

Continuing I/O configuration changes without reIPL

If the failure does not require a reIPL, and you want to continue making dynamic configuration changes from the point of failure, enter the ACTIVATE command with the RECOVER keyword and TARGET option. z/OS continues making the remaining dynamic configuration changes, finishing building the new (target) I/O hardware configuration. If you do not specify TARGET or SOURCE option, z/OS continues in the same direction as when failure occurred.

Recovering original I/O configuration with reIPL

If the system failure requires a reIPL, and you want to recover the I/O configuration that was active before the failure, follow these steps:

1. Specify the source IODF in the LOADxx member of SYS1.PARMLIB or SYSn.IPLPARM. For more information, see “IODF processing at IPL” on page 76.
2. ReIPL the system.
3. Enter the ACTIVATE command with the RECOVER keyword and SOURCE option. z/OS backs out all dynamic hardware changes made prior to the failure, returning the system to its original (source) I/O hardware configuration.

Continuing I/O configuration changes with reIPL

If the failure requires a reIPL, and you want to continue making dynamic configuration changes from the point of failure, follow these steps:

1. Specify the target IODF in the LOADxx member of SYS1.PARMLIB or SYSn.IPLPARM. For more information, see “IODF processing at IPL” on page 76.
2. ReIPL the system.
3. Enter the ACTIVATE command with the RECOVER keyword and TARGET option. z/OS continues making the remaining dynamic changes, finishing building the new (target) I/O hardware configuration.

Recovering LPAR mode dynamic configuration changes

In LPAR mode, after a failure during a dynamic configuration change, you can recover the system to its original I/O configuration or continue making dynamic configuration changes from either the logical partition that had failed or from any other logical partition.

Recovering original LPAR mode I/O configuration without system reset

If your processor was running in LPAR mode, the failure does not require a system reset, and you want to recover the I/O configuration that was active before the failure occurred, follow these steps:

1. In all logical partitions, to make software-only changes, activate the source IODF.
2. In any logical partition, enter the ACTIVATE command with the RECOVER keyword and SOURCE option. z/OS backs out all dynamic configuration

changes made prior to the failure, returning the system to its original (source) I/O hardware configuration, which is shared by all logical partitions.

Continuing LPAR mode I/O configuration changes without system reset

If your processor was running in LPAR mode, the failure does not require a system reset, and you want to continue making dynamic configuration changes from the point of failure, do the following:

- In any logical partition, enter the ACTIVATE command with the RECOVER keyword and TARGET option. z/OS continues making the remaining dynamic configuration changes, finishing building the new (target) I/O hardware configuration, which is shared by all logical partitions.

Recovering original LPAR mode I/O configuration with system reset

If your processor was running in LPAR mode, the failure requires a system reset, and you want to recover the I/O configuration that was active before the failure occurred, follow these steps:

1. In the logical partition that failed, specify the source IODF in the LOADxx member of SYS1.PARMLIB or SYSn.IPLPARM. For more information, see "IODF processing at IPL" on page 76.
2. For Enterprise System/9000, perform system reset on the logical partition that failed. For more information, see *Enterprise System/9000 Operator Guide*.
3. In all logical partitions, make software-only changes, activating the source IODF.
4. In any logical partition, enter the ACTIVATE command with the RECOVER keyword and SOURCE option. z/OS backs out all dynamic configuration changes made prior to the failure, returning the system to its original (source) I/O hardware configuration, which is shared by all logical partitions.

Continuing LPAR mode I/O configuration changes with system reset

If your processor was running in LPAR mode, the failure requires a system reset, and you want to continue making dynamic configuration changes from the point of failure, follow these steps:

1. In the logical partition that failed, specify the target IODF in the LOADxx member of SYS1.PARMLIB or SYSn.IPLPARM. For more information, see "IODF processing at IPL" on page 76.
2. Perform system reset on the logical partition that failed. For more information, see *Enterprise System/9000 Operator Guide* .
3. In any logical partition, enter the ACTIVATE command with the RECOVER keyword and TARGET option. This continues making the remaining dynamic configuration changes, finishing building the new (target) I/O hardware configuration, which is shared by all logical partitions.

Using HCM to migrate to IBM zSeries z890 or z990

With z/OS V1R4 and z/VM V4R4, a new FMID HCM1510 is provided for HCM. If you have installed this FMID or a later version of HCM, then HCM can handle the IBM zSeries 990 (z990) and the IBM zSeries 890 (z890). You must also have FMID HCS7708 for HCD installed - HCM needs this FMID to provide support for the new server.

With z/OS V1R7, all toleration PTFs must be installed before HCM can use the z9109 platform.

This topic describes scenarios how to migrate from an I/O configuration attached to a pre-z990 processor to a z990 or z890 processor using HCM. If you are using HCD to migrate to z990 or z890 processors, use *z/OS HCD User's Guide*.

The major difference between the new z9109, z990 and z890 processors and their predecessors is that the new processor definitions contain logical channel subsystems. Therefore, a new object has been introduced into the hierarchy for a processor definition:

```
Processors –> Channel subsystems –> Partitions –> Channel paths
```

With this new hierarchy you cannot simply edit the processor type/model information within the HCM or HCD dialog, but you need to perform a series of migration steps using HCM and HCD. These steps are described in the following.

This topic discusses the following alternatives:
- "Configurations with logical definitions only"
- "Configurations including physical definitions" on page 108

Note:
Before you start the migration steps, we recommend that you first read the topic on importing and exporting data in *z/OS and z/VM HCM User's Guide*. This topic describes the steps for exporting or importing data, and also explains what information you can update when importing files.

Configurations with logical definitions only

If you use HCM for your configuration definitions without documenting your physical environment, then migrating from a pre-z990 processor to a z990 or z890 processor is a straightforward task. Perform the following steps in the given order:

1. Create backup copies of your HCM configuration file and of your IODF.
2. Create a new z990 or z890 processor using HCM, and define the required number of logical channel subsystems.
3. Exit HCM, and continue to work on the IODF using HCD:
   - Navigate to the **Processor List**:
     a. Look for the pre-z990 processor with the I/O configuration that you want to migrate. The basic migration task is to copy the I/O configuration of the pre-z990 processor into a logical channel subsystem on the new

processor. Therefore, on your source processor, apply action ***Copy to channel subsystem. . . (SMP)*** from the context menu (or action code y).

   b. Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
   c. Specify the target processor ID and select the logical channel subsystem ID as requested. Pressing the Enter key copies the source processor I/O definition into the logical channel subsystem of the target processor. Note that coupling facility connections are not copied.

- Delete the source processor definition that you copied into the logical channel subsystem.
- Establish the coupling facility connections between the CF CHPIDs on other processors and the CHPIDs in the logical channel subsystem into which you copied the processor definition. Also establish the CF connections within the z990/z890 processor (if needed). Note that you first might need to set the CF CHPIDs in the target logical channel subsystem to 'not occupied'.
- Perform the PCHID migration for the CHPIDs on the new z990 processor using HCD and the CHPID Mapping Tool as described in the *z/OS HCD User's Guide*.

4. Exit HCD and return to HCM. Reload the HCM configuration file. HCM will now force a resynchronization with the IODF. Check whether your CTC connections from and to the logical channel subsystem are as expected. You need not check CTC connections between processors not involved in the migration.

   HCM detects and establishes all new switched CTC connections, but does not recognize point-to-point CTC connections. Establish these point-to-point CTC connections using the ***Create CTC Connections dialog***. You need not type the CTC control units and CTC devices into the dialog, because HCM detects that the selected CHPIDs are already connected to CTC control units and CTC devices. Simply accept HCM's proposed CTC control units and CTC devices, and the point-to-point CTC connections will also be documented in HCM.

   Note that the CTC connections themselves were already logically defined in the IODF, but HCM has now also documented their physical configuration.

5. The migration process is now complete. The pre-z990 processor I/O configuration has been successfully copied into a z990/z890 processor's logical channel subsystem definition.

Configurations including physical definitions

If you use HCM to document both your logical and physical environment, then the migration task consists of a few more steps. Perform the following steps without changing the order:

1. Create backup copies of your HCM configuration file and of your IODF.
2. **Generic Converters:** If there are any connections defined between CHPIDs of the source processor definition and other objects where a converter is involved, then make sure that the converter is not a 'generic' converter that was automatically created by HCM. Double click on the converter object, and press the **Replace** button. On the following **Replace Converter** dialog, create a converter with label and serial number — you now have your own converter in the connection. If you keep generic converters, you will receive error messages during the later migration steps. If you do not want to create your own converters, but want to keep generic converters only, then you need to disconnect all connections to the processor that is being migrated where converters are involved in the link. Remember to reconnect them at the end of the whole procedure.

3. Save the HCM configuration file, and export the HCM files 'CONNECT.txt' and 'LOGLINK.txt'.

   You might receive some error messages during the export step. If you receive error message

   ```
   'EEQ0261E: CCI<controller interface> is not unique,
   so it has not been exported'
   ```

   then check whether the mentioned interface is directly connected to a CHPID of the source processor. If so, make sure that the controller interface is unique in your configuration. Most likely, you have created your own PDF file and labeled different interfaces with the same name. Make sure that your interfaces are uniquely labeled in the PDF files.

   In case you receive message `'EEQ0261E: CCI ...'` or `'EEQ0261E: CDI ...'`, check whether one of your PDF files is set up in a way that two or more interfaces have the same label. If this is the case, copy and modify the PDF file to use unique labels for all interfaces. For the modified PDF file, use a different file name (to preserve the original PDF file if needed), and provide a new subsystem identifier within the modified PDF file. After you modified the PDF file, you need to change the physical description for all controllers where you used the original PDF file, and you now need to apply the new PDF file. You can do so by clicking on the ***Edit Controller*** dialog where you then can select your modified PDF file.

   In contrast to normal processing, HCM cannot handle duplicate objects like interface names as soon as data gets exported. So it is important that all your objects and all your interfaces are uniquely labeled.
4. Create a new z990/z890 processor using HCM, and define the required number of logical channel subsystems.
5. Exit HCM, and continue to work on the IODF using HCD:
   - Navigate to the **Processor List**:
     a. Look for the pre-z990 processor with the I/O configuration that you want to migrate. The basic migration task is to copy the I/O configuration of the pre-z990 processor into a logical channel subsystem of the new processor. Therefore, on your source processor, you apply action ***Copy to channel subsystem. . . (SMP)*** from the context menu (or action code **y**).
     b. Specify the IODF to which the selected operating system is to be copied. The default IODF is the IODF you are currently working with.
     c. Specify the target processor ID and select the logical channel subsystem ID. Pressing the Enter key copies the source processor I/O definition into the logical channel subsystem of the target processor. Note that coupling facility connections are not copied.
   - Delete the source processor definition that you copied into the logical channel subsystem.
   - Establish the coupling facility connections between the CF CHPIDs on other processors and the CHPIDs in the logical channel subsystem, and also establish the CF connections within the z990/z890 processor (if needed). Note that you first might need to set the CF CHPIDs in the target logical channel subsystem to 'not occupied'.
   - Perform the PCHID migration for the CHPIDs on the new z990 processor using HCD and the CHPID Mapping Tool as described in the *z/OS HCD User's Guide*.
6. Exit HCD and return to HCM.
7. Use a text editor on your PC to edit the two files that were exported in step 3. In 'LOGLINK.txt' and 'CONNECT.txt', globally edit every occurrence of the old processor name. Change the old name to the new z990 processor name

followed by a dot (.) and the logical channel subsystem ID. For example: exchange 'PROC2064' with 'PROC2084.1' if you have copied a processor named PROC2064 into logical channel subsystem 1 of processor PROC2084.

However, before you globally replace the old processor name, make sure that there are no other occurrences of the processor name, for example in a string label or a patchport label.

8. Use HCM and log on to the host again. Do not load the HCM configuration file, but instead, execute the Import data... action from the File menu. This allows you to import the modified files into your HCM configuration file that you want to update. During the import step, HCM asks you whether you want to use the same IODF that is already associated with the HCM configuration file. Answer YES to select that option. HCM now performs a background resynchronization step while importing the data. During the import step, you may receive several HCM error messages:
   - Error message **EEQ0251I** may appear saying "Record not processed. The configuration contains more than one PATCHPRT object which matches 'xxxxxx'.". This indicates that you have patchports in your configuration with patchport labels that have been manually entered and are identical to those of other patchports. You can avoid such messages by specifying patchport labels that are unique across the whole configuration and are not default labels generated by HCM for other patchports.
   - Error message **EEQ0251I** may appear saying "Record not processed. A STRING object with the identifier 'String #n' does not exist.". String objects are never directly connected to CHPIDs, so you need not expect problems from this point of view. The record was not processed as reported, so the current physical environment was not changed.
   - Error message **EEQ0251I** may appear saying for example: "Record not processed. A CONTRLLR object with the identifier 'Controller for PROC2084.1' does not exist.". In this example, we are migrating from processor PROC2064 to logical channel subsystem 1 of processor PROC2084. If the original label of the controller was 'Controller for PROC2064', and if you performed a global change of the processor name even though the processor name 'PROC2064' is part of at least one label of another object, then this error message occurs. This error message may also involve further error messages as a consequence. Check whether in step 7 on page 109 you performed an unintentional name change in files 'LOGLINK.txt' and 'CONNECT.txt'.
   - Warning message **EEQ0253I** may appear saying that an object has been disconnected from a CHPID of the deleted processor definition. These are disconnect requests for HCM physical connections only; no IODF update is involved with this disconnect request. Note that the I/O configuration below the deleted CHPID remains persistent for the new CHPID on the logical channel subsystem. For example, if the deleted CHPID was connected to patchport 1, which in return was connected to patchport 2, and patchport 2 was finally connected to a switch port, then both patchports will be part of the fiber link configuration between the switch port and the new CHPID on the logical channel subsystem. Additionally, the cables involved in this link also remain in their original position.
   - HCM does not write certain links into the exported file 'CONNECT.txt'. The missing connections relate only to daisy-chained controller connections where the first controller interface is further connected via a converter. When you perform the import step 8, HCM detects that a link is missing. In this case, you will receive several EEQ0251E error messages saying that records have not been processed followed by one of these messages:
     – This connection requires another CONNECT record that references the bottom of converter "MyConverter".

   - This connection cannot be made because "3480 #1.A.out" is not connected.
   - This connection cannot be made because the connection in file C:\...\CONNECT.txt at line #000011 could not be made.
   - 'MyConverter' is not a valid LOGLINK Target.

   A fix is being developed and will be made available via a new service level. Check the HCD/HCM home page to find out when this HCM service level is available.

   A circumvention for this problem is to disconnect the connection between the first controller interface of the daisy-chained controller interfaces from the logical target (the logical target is not the converter because the converter is just inserted in the link for reasons of documenting the physical configuration). Because the critical link is no longer in the configuration, you could now go on as if you had disconnected all connections where converters were involved. See step 2 on page 108.

9. After the import/resynchronization step, you need to check whether your CTC connections from and to the logical channel subsystem are as expected. You need not check CTC connections between processors that are not involved in the migration.

   HCM has detected all new switched CTC connections, but cannot recognize point-to-point CTC connections. Establish those point-to-point CTC connections via the ***Create CTC Connections*** dialog. You need not type the CTC control units and CTC devices into the dialog, because HCM detects that the selected CHPIDs are already connected to CTC control units and CTC devices. Simply accept HCM's proposed CTC control units and CTC devices, and the point-to-point CTC connections will also be documented in HCM. Note that the CTC connections themselves were already logically defined in the IODF, but HCM has now also documented their physical configuration.

10. The migration process is now complete. The pre-z990 processor I/O configuration has been successfully copied into a z990/z890 processor's logical channel subsystem definition.

Migrating to the dynamic capability provided with HCD

For installations that use MVS configuration program (MVSCP), this section describes how to migrate to the dynamic configuration capability provided in MVS/ESA SP 4.2 and later.

For a detailed migration procedure for migrating to HCD, see *z/OS HCD User's Guide*.

Before you can use dynamic configuration in your installation, you must make these changes:

1. Convert the installation programs that perform UCB look-ups, scan UCBs, use unit verification services, and are sensitive to I/O configurations. See "UCB services considerations."
2. Change the default of static to dynamic for any devices that you want to be able to change or delete from the software definition. (This change is not required to dynamically add a new device definition.) See "Specifying devices as dynamic after a migration" on page 116 for information about changing to dynamic devices that defaulted to static at migration.

If your installation has not prepared or is not going to use dynamic configuration, you must define devices that support dynamic configuration as installation-static.

Note: Installations that IPL with I/O definition files (IODFs) that define dynamic devices should use the IPCS subcommands FINDUCB, LISTUCB, and IOSCHECK to validate and display dynamic UCBs. Any other subcommands might not produce valid results. FINDUCB, LISTUCB, and IOSCHECK are also the subcommands to use for validating and displaying UCBs for devices that have changed from installation-static to dynamic through an ACTIVATE command.

UCB services considerations

There are considerations for accessing UCBs when you migrate to dynamic capability.

Converting references to UCB services for dynamic

If you wish to use dynamic configuration, and your installation has programs that obtain unit control block (UCB) addresses or use unit verification services, you must convert the programs to use services that support dynamic configuration. Also, you must understand the environment in which the services work.

Some old UCB services do not support four digit device numbers, UCBs that reside above 16 megabytes, or dynamic devices. These services are known as limited method UCB services:

- The IOSLOOK macro
- The UCB scan service (address in CVTUCBSC)
- The allocation unit verification service

As described in “Specifying an I/O device number” on page 42, a device number can be any hexadecimal number from X'0000' to X'FFFF'. You can use the limited method services to access only UCBs for devices whose device number is in the range X'0000' to X'0FFF'.

As described in “Defining an above 16 megabyte UCB” on page 53, you can use HCD to define a device's UCB above 16 megabytes if the device's unit information module (UIM) indicates that the device support code enables UCBs above 16 megabytes. You can use the limited method services to access only UCBs that reside below 16 megabytes (that is in 24-bit addressable storage).

As described in “Defining whether an I/O device will be dynamic” on page 46, devices can be defined as one of three different types:

- **Static**: cannot add, change, or delete a device to the software definition dynamically
- **Installation-Static**: can add, but cannot change or delete a device to the software definition dynamically
- **Dynamic**: can add, change, or delete a device to the software definition dynamically

If you define a device as static or installation-static, the limited method services can be used. If you define the device as installation-static, you can still dynamically add its UCB, making it accessible to your programs that use limited method UCB services. If you do not need to dynamically change or delete the UCB of a device that supports dynamic configuration, you can define that device as installation-static, and you do not need to change any applications.

If you want to have devices defined as dynamic, you must convert references to limited method UCB services to general method UCB services. If you have any application programs or products that need conversion, do not define devices as dynamic until you have converted the programs or installed the converted products. An application program that has not been converted cannot recognize devices defined as dynamic. To the application program, it appears as though the device is not defined to z/OS.

Note: You do not need to change applications that access UCBs through TIOT or DEB fields.

To enable your programs to use and recognize UCBs that reside above 16 megabytes, or define dynamic devices or devices that have four digit device numbers, you need to replace:

- The IOSLOOK macro with the UCBLOOK macro.
- The UCB scan service (address in CVTUCBSC) with the UCBSCAN macro.
- The allocation unit verification service with the EDTINFO macro.

You also need to follow the new serialization requirements when using UCB services to obtain UCB addresses as described in “Consideration when accessing UCB data” on page 115. Table 10 on page 115 lists the UCB services that replace services that do not support dynamic configuration. For more information on programming requirements for the services, see:

- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *z/OS MVS Programming: Assembler Services Guide*

Table 10. Description of UCB Services

Configuration Service	Description
UCBLOOK	Obtains the UCB common segment address of a given device number or a given volume serial number.
UCBSCAN	Scans the UCBs.
UCBPIN	Prevents a UCB for a device and its associated device control blocks from being dynamically deleted (called pinning the UCB); allows the UCB and associated device control blocks to be unpinned when the program is finished using it.
IOCINFO	Returns the I/O configuration token that indicates which configuration definition is active on z/OS. The token can be used as input for other services to synchronously detect configuration changes.
CONFCHG	Requests notification of a dynamic configuration change.
EDTINFO	Obtains information from the EDT.

Consideration when accessing UCB data

When you use the UCB services to obtain UCB addresses, you need to ensure that the device is not dynamically deleted while you are obtaining information, because the UCB for the device and the related data structures would be deleted. For authorized programs, you can prevent the UCB from being deleted by pinning the UCB. See *z/OS MVS Programming: Authorized Assembler Services Guide* for information about pinning.

For unauthorized (and authorized) programs, you can prevent a UCB from being deleted by allocating the device. Also, for unauthorized programs, if you obtain a copy of the UCB, a deletion of the original UCB does not affect the copy. You can obtain copies of a UCB through the UCBSCAN macro.

Note: It becomes the application's responsibility to detect when the real UCB has changed and the copy has become obsolete. This detection can be accomplished synchronously by using the configuration token with the UCB services or asynchronously with the CONFCHNG exits.

Authorized services with UCBs as input require the address of a UCB not the address of a copy of the UCB. Many unauthorized services with UCBs as input do not accept addresses of UCB copies either. However they require that the devices be allocated which prevents the UCB from being deleted while in use anyway. You can pass addresses of UCB copies as input only to those unauthorized services listed in “Services accepting a UCB copy” on page 116.

To obtain UCB copies, you use the UCBSCAN macro with the COPY keyword. With the COPY keyword, you also need at least one other keyword. You can use either of the following keyword combinations:

- COPY and UCBAREA - The UCBAREA keyword specifies the address of a workarea that receives the UCB copy.
- COPY and UCBAREA and CMXTAREA and DCEAREA - The CMXTAREA and DCEAREA keywords specify the addresses of storage areas to receive tokens with the copy.

Services accepting a UCB copy

Some services accept copies and some accept copies but require tokens (from CMXTAREA and DCEAREA keywords) as part of the copies. The unauthorized services that accept copies but do **not** require tokens as part of the copy are:
- LSPACE
- TRKCALC

The unauthorized services that accept copies and require tokens as part of the copy are:
- CVAFTST
- DEVTYPE

Note: You can use these unauthorized services with UCB copies only with DFSMS.

Use macro IHADFA to check the DFSMS level. CVTDFA points to DFA, which is mapped by macro IHADFA. The DFAREL field contains a two-byte release level. The previously listed services are available only if DFAREL is equal to or greater than X'3300'. (You can also use a callable service to check the DFSMS release level; see *z/OS DFSMSdfp Advanced Services*.)

Specifying devices as dynamic after a migration

During migration from MVSCP to HCD, a work IODF is created. How devices are defined in this IODF is a consideration for using dynamic capability. Also when you convert an IODF from MVS/ESA SP 4.1, the same consideration is present. An IODF in these two scenarios, defines all devices as static by default. That means that if the device has support code for dynamic capability the device is installation-static. An installation-static device cannot be changed or deleted dynamically in the software definition. Therefore you will probably want to modify the IODF to override its installation-static default characteristics for specific devices.

The scope of modification is installation-specific, dependent on installed IBM products, other vendor products, and installation applications that are sensitive to I/O configuration changes. The scope of modification also depends on the abilities of these products and applications to support and coexist in the new environment.

Use HCD to redefine the devices. For information about how to change device definitions, see *z/OS HCD User's Guide*.

Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you may view the information through the z/OS Internet Library Web site or the z/OS Information Center. If you continue to experience problems, send an e-mail to mhvrcfs@us.ibm.com or write to:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer or Library Server versions of z/OS books in the Internet library at:

`http://www.ibm.com/systems/z/os/zos/bkserv/`

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Index

Program Number: 5694-A01

Printed in USA

GA22-7525-14