z/OS and z/VM

IBM
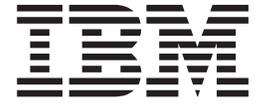
Hardware Configuration Manager User's Guide

z/OS and z/VM

Hardware Configuration Manager User's Guide

IBM

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page 435.

Fifth Edition, September 2004

This edition applies to Version 1 Release 6 of z/OS (5694-A01), Version 1 Release 6 of z/OS.e (5655-G52) , Version 5 Release 1 of z/VM® (5741-A05) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM® representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

    IBM Deutschland Entwicklung GmbH
    Department 3248
    Schoenaicher Strasse 220
    D-71032 Boeblingen
    Federal Republic of Germany

Or by FAX:
    From Germany: 07031-16-3456
    Other countries: +(49)+7031-16-3456

Or electronically:
    Internet e-mail: s390id@de.ibm.com

> **Internet**
>
> Visit our homepage at http://www.ibm.com/servers/eserver/zseries/zos/hcm/

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Contents   **v**

Figures

Tables

About this document

This guide explains how to use **Hardware Configuration Manager** (**HCM**), a comprehensive, graphical hardware configuration and operations tool available for IBM and compatible S/390® systems. HCM allows system administrators to easily and efficiently manage all information about the hardware and connectivity of their systems.

Who this document is for

The *HCM User's Guide* explains to system administrators, system programmers, system operators, system planners, and your hardware support staff how to install and use HCM to create and maintain hardware configuration information.

It is assumed that the person who is responsible for defining and managing hardware configuration for a z/OS or z/OS.e or a z/VM system has basic knowledge of z/OS and z/VM and hardware configuration.

For information on configuration planning for z/OS or z/OS.e, refer to *z/OS HCD Planning*. For information on configuration planning for z/VM, refer to *z/VM: I/O Configuration*.

How this document is organized

This document contains the following chapters:

Chapter 1, "Overview," on page 1
> summarizes HCM's major features and benefits.

Chapter 2, "Installing HCM," on page 3
> explains how to install the software to be used for z/OS and z/VM and how to establish the host communication for both operating systems.

Chapter 3, "Getting started," on page 39
> explains how to use the configuration diagram, dialogs, menus and reports, as well as the mouse and keyboard shortcuts in HCM.

Chapter 4, "Building your configuration," on page 63
> describes the relationship between HCM and HCD (IODFs, config files); explains physical and logical objects; provides information on how to use HCM for the first time (creating a new configuration from an IODF or from scratch), how to open an existing configuration and how to save, close or copy the configuration.

Chapter 5, "Standalone mode," on page 91
> explains how to invoke and work with the stand-alone mode.

Chapter 6, "Modifying objects in your configuration," on page 97
> explains how to manipulate all the elements of your hardware configuration with the interactive configuration diagram and dialogs: processors, partitions, CHPIDs, controllers, control units, strings, units, devices, converters, directors, crossbar switches, cabinets and patchports.

Chapter 7, "Connecting objects in your configuration," on page 155
> explains how to connect and define logical objects.

Appendix G, "Special setup considerations for z/OS," on page 429
discusses special considerations related to setup.

Appendix H. Accessibility
discusses z/OS product accessibility features that help users who have a physical disability to use the software.

Related information

For an overview of the documentation associated with z/OS, see the *z/OS Information Roadmap*, SA22-7500.

Please see also our HCM home page

http://www.ibm.com/servers/eserver/zseries/zos/hcm/

which provides information concerning product updates, newsletters, conferences and more. For more information on **z/OSHCD**, see *z/OS HCD User's Guide*.

If you want to use HCM for defining I/O configurations on z/VM systems, the following documentation associated with z/VM or other products may be useful:

Table 1. Related z/VM I/O Configuration References

Title	Order Number
z/VM: I/O Configuration This book contains detailed information on z/VM HCD.	SC24–6044
z/VM CP Command and Utility Reference	SC24–5967
z/VM CP Planning and Administration	SC24–6043
z/VM: System Messages and Codes – CP	GC24–6030
z/VM Diagnosis Guide	GC24–6039
z/OS and z/VM HCD Messages	SC33–7986
z/OS MVS Device Validation Support	GA22-7525
IOCP User's Guide and ESCON Channel–to–Channel Reference	GC38–0401
Input/Output Configuration Program User's Guide for IYP IOCP	SB10–7029
Input/Output Configuration Program User's Guide for ICP IOCP	SB10–7037
ES/9000® Processor Complex Operator Guide	SC38–0085

The latest editions of most z/VM publications are available in Adobe Portable Document Format (PDF) from the z/VM Internet Library:
http://www.ibm.com/eserver/zseries/zvm/library

Service levels

From time to time, IBM makes available service levels to update HCM. These service levels are numbered in ascending order according to their release date. A new service level comprises new updates as well as all updates reflected in earlier service levels. That is, you only have to install the newest service level to obtain all product updates. Always see the EEQREAD.ME file before you use the new service level. It resides in the product directory after you have installed the service level.

For information on service levels, see the HCM home page:
http://www.ibm.com/servers/eserver/zseries/zos/hcm/

Summary of changes

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.

The following sections list the changes that have been made to HCM with various operating system releases.

What is new in HCM for z/OS V1R6 and z/VM V5R1

Summary of changes for SC33–7989–05 issued September 2004

Queue prioritization on OSD channels

OSD channels support four different priorities for I/O queues. However, if users decide to not assign different priorities to the I/O queues, then the OSD channels can support four times as many queues as with priorities defined (4 * 480 = 1920). When defining or modifying OSD channels, HCM offers an option to enable or disable queue prioritizing.

Support of the CHPID Mapping Tool in HCM

HCM can interact with the CHPID Mapping Tool in order to add or update PCHIDs to a configuration. Users can invoke a new utility called *CHPID Mapping Tool Support*. This utility offers two functions:

- It exports and downloads an IOCP file of a selected processor from the host. With this IOCP file as input, you can now invoke the CHPID Mapping Tool. The CHPID Mapping Tool assigns the PCHID values to the CHPIDs in the IOCP file and saves the modified file on the workstation.

- It uploads the modified IOCP file to the host, which now contains the new or updated PCHID values. On the host, the utility automatically invokes the PCHID migration into the IODF. Then it also refreshes the local configuration on the workstation with the new CHPID-PCHID assignment.

Enhanced configuration diagram

In the configuration diagram, spanned CHPIDs are shown with one instance in each channel subsystem over which they are spanned. However, HCM ensures that you can physically connect only one of these instances to a target. The configuration diagram shows which of the instances is physically connected and helps you to navigate to this instance from any other instance associated with the same spanned CHPID.

What is new in HCM for z/OS V1R5 and z/VM V5R1

Summary of changes for SC33–7989–04 issued May 2004

With the appropriate PTFs installed, you can take advantage of the following new functions in HCM:

Support of spanned channel paths

Spanned channels are supported in the following aspects:

- You can connect an object's interface (for example, a controller interface) directly to a spanned CHPID. This handling is available with the *Select Connection Target(s)* and *Select Multiple Connection Target(s)* dialogs.
- You can logically and physically disconnect object interfaces from spanned CHPIDs.
- When locating CHPIDs, CHPID connections or cables in the configuration diagram, all instances of a spanned CHPID are included.

Support of dynamic LPARs

For IBM @server zSeries® 990 and zSeries 890 processors with the appropriate hardware and software support levels, HCM provides the capability to add or remove reserved LPARs into or from the configuration. You can use HCD for the dynamic I/O configuration of reserved partitions after a POR. You can define partitions with a placeholder ID ('*') and change the '*' to a partition name later or vice versa. The '*' indicates that you plan to add a partition or to remove an existing partition dynamically by activating the configuration at a later point in time.

Over-defining CHPIDs on an XMP processor

For an XMP processor, you can define a channel path that is not physically installed on the machine. This may be useful if you want to migrate from a machine which had more channels defined than the target XMP processor has currently installed, or if you want to prepare a configuration for future upgrades of the channel cards.

Support of null device candidate lists for XMP processors

If devices are connected to a control unit which is shared between multiple channel subsystems, some (not all) of these devices may specify a null device candidate list for one or more CSSs.

More user fields

HCM now allows you to define twenty user data fields to characterize a hardware object. In addition to specifying an ID, description, serial number or type, you can now record further information such as the lease status, corporate asset number, or physical location (building and/or data center map grid) for a given piece of hardware.

Select multiple connection targets

A new dialog called **Select Multiple Connection Target(s)** is available for controllers and switches. With this dialog, you can connect multiple controller interfaces or switch ports to multiple connection targets in one step, while in previous releases, you had to select the connection targets iteratively.

Enhanced views of the configuration diagram

A second method to crop the configuration diagram is now available: HCM allows you to crop the diagram according to one or more previously selected object(s). Thus you can restrict the view to the selected objects and their connections to other objects. For more information, see "Keyboard" on page 41.

Also, you can use F12 to display the previously saved view of a configuration diagram.

New functions for saving information

Pressing the corresponding **Save** button in the following dialogs lets you store the displayed information:

- **Save Port List** button in the **Switch** dialog: The information displayed in the visible columns of the **Ports** section is stored in the specified file.

- **Save As** button in **List Devices for OS Configuration:** With this button you can store all information pertaining to the selected operating system in the specified file.

What's new in HCM for z/OS V1R4 and z/VM V4R4

With the availability of z/VM HCD, users can now use HCM as the graphical and interactive interface for defining hardware and software configurations also on z/VM systems, just as on z/OS systems. The information on how to set up and customize HCM for z/VM is new in this edition. This information and all other information related especially to z/VM is indicated by an '*' on the left margin, while new and changed information related to z/OS is marked with a '|'.

Support of multiple logical channel subsystems (LCSS)

A note on terminology:

Throughout this document, the following terms are used:

XMP processor and SMP processor
> The term **XMP processor** designates processors that support multiple logical channel subsystems. It is used in contrast to the term **SMP processor**, which designates processors of previous generations that support only one channel subsystem.

XMP processors support the definition of multiple logical channel subsystems in which partitions and CHPIDs are defined. Thus, more than 256 channel paths are supported. The number of definable logical channel subsystems depends on the processor.

With XMP processors, some types of channel paths can be shared across partitions belonging to different channel subsystems. These channel paths are called *spanned CHPIDs*. You find information on how to define multiple logical channel subsystems and spanned channel paths in "Processors" on page 97 and "CHPIDs" on page 103.

Support of physical channel identifiers

Real I/O hardware is attached to a processor via physical channels. For XMP processors, the physical channels need a physical channel identifier (PCHID) which determines the physical location of a channel in the processor. For these processors, you have to specify the physical channel identifier (PCHID) related to the channel path identifier (CHPID).

The task of adding the physical channel path information to an IODF is eased in a way that HCD can cooperate with the CHPID Mapping Tool (CMT). Input to the CMT is the hardware configuration file of your machine (CFReport, see also in the "Glossary" on page 439) and a valid IOCP input file. Output from the CMT is again an IOCP input file that has the PCHID values filled in. This IOCP input file

can be reimported into the IODF. For more information on the CHPID Mapping Tool refer to the applicable *z/OS HCD User's Guide* or *z/VM: I/O Configuration* book.

Other changes

The *IODFs* dialog now shows the size of the IODF and

- for z/OS, the creation date
- for z/VM, the date of the last update.

What's new in HCM for z/OS V1R3

> **Note**
>
> This release of z/OS does not include a new release of HCM. HCM Release 2.9.0 remains the current release.

Cascaded switch support

Cascaded connections between FICON® switches are supported. This means that a control unit to a FICON channel path connection can now run through two FICON switches. Therefore, a switch address must be specified for the single FICON switches of the fabric. HCM also supports explicit upgrading from one–byte link addresses to two–byte link addresses as well as down–grading of two–byte link addresses to one–byte link addresses via a new utility.

FCTC (FICON channel-to-channel) support

Two FC channels might be connected with each other to establish an FCTC connection. At least one FC channel path must be connected to a 2064 processor type in order to provide a valid FCTC connection. HCM together with HCD ensures that this rule is obeyed.

Support for iQDIO channel paths

HCM allows you to specify the maximum frame size for IQD channel paths. HCM's reports have been enhanced so that they contain the maximum frame size (where appropriate).

Support for PPRC (peer to peer remote copy)

HCM's support consists of the following elements:

- Document direct physical connections between serial interfaces on controllers. These connections may pass through directors and patchports.
- Maintain logical connections between subsystems, including through directors.
- Ability to share PPRC data with other tools (like GDPS® and RCMF). HCM can export and import CESTPATH commands and GEOPARM control statements.

In order to create a PPRC connection, HCM offers you a wizard where you create a PPRC connection step by step. You can add an SSID (Subsystem Identifier) to control units, and selected physical description files have been updated to include SAID (System Adapter Identifier) information. See also "Connecting controllers to controllers (PPRC connections)" on page 171.

Other changes

Other improvements are as follows:

- In the *Preferences* menu (selectable via *View*), you can now switch between vertical and horizontal lines as well as diagonal lines for displaying connections between objects in the diagram. HCM also prints these diagonal lines if they are selected. Together with the option *Compress diagram horizontally* found in the *Preferences* dialog, the number of pages that HCM needs for printing often gets reduced.
- Both the *Select Connection Target(s)* and the *Edit Fiber Link* dialogs have been updated to display a cabinet's description.
- The following dialogs have been enlarged to display more information:
  - Edit String
  - Controller
  - Control unit <=> Processor Definitions
  - Control Unit List
  - Switch
  - Select Connection Target(s) from patchports
  - Select CHPID for CTC Connection
  - Crossbar Switch
  - Edit Crossbar Control Unit Interface
- Support for a number of devices column in the *Edit/Locate CHPID* dialog has been introduced. Also, the Processor Summary and Channel Path Connections reports display this column.
- The message text for disconnecting a patchport that has been permanently connected is improved. The message now reads: *"%s may not be removed from its fiber link because %s is permanently connected all the way to %s. Use the Connection is Permanent option on the Assign Cable dialog to change the connection type."*
- The Cabinet reports have been enhanced. It is now recommended to print this as a separate report using Landscape. The number of columns is calculated based on paper size.

What's new in HCM for z/OS V1R1

> **Note**
>
> This release of z/OS does not include a new release of HCM. HCM Release 2.9.0 remains the current release.

Dynamic Channel Path Management (DCM)

As part of the z/OS Intelligent Resource Director, Dynamic Channel Path Management allows the system to dynamically re-assign channel paths to connected control units based on the current work load and its service goals. HCM allows channel paths to be designated as static (fixed) or managed (re-assignable) when they are defined.

New utility to prime IODF with sensed data

See "Priming the IODF with sensed data" on page 204.

Other changes

Other improvements are as follows:

- **Change of filename extension for Physical Description Files.**
  The filename extension for physical description files was changed from ".pdf" to

".txt" for HCM 2.9 Service Level 3 and higher. Nevertheless, HCM 2.9 Service Level 3 and higher can still read physical description files having the extension ".pdf".

- **Selective cabinet reports.**
  The Front View and Back View of patchports can be printed separately.

- **Loading HCM configuration files.**
  A HCM configuration file (either *.hcm or *.hcr) will automatically be loaded with HCM if you double-click with the left mouse button on the HCM configuration file in the Windows® Explorer, for example.

- **Processor and CHPID Connections dialog.**
  The **Reachable CHPIDs** dialog has been replaced by the **Processor and CHPID Connections** dialog in order to provide a better overview of which channel paths the object under focus will connect with. The new dialog contains a tab for each processor, where all CHPIDs reachable from that processor are listed. Additionally, the new dialog allows specifying the number of managed CHPIDs of each processor to which the object of interest should be connected. A summary tab shows the processor definitions for the object under focus, so that you can easily verify that all connections you want to establish will be defined after pressing the 'OK' button.

What's new in HCM for OS/390® V2R9

Physical connectivity enhancements

HCM has been enhanced to allow the following:

- Permanent and/or anonymous trunk connections can be made between patchports and director ports.
- Connection changes are possible without having to rebuild all or most of the ESCON link information.
- Infrastructure connections can be established after an IODF load.
- Multiple patchports can be disconnected simultaneously.

As a result of these enhancements, HCM will now differentiate serial connections as "permanent" or "mutable."

The idea is that infrastructure connections such as trunk cables from processors to cabinets can be marked as permanent so that HCM can prevent them from being disconnected during normal connect/disconnect procedures. Other connections can be marked as "mutable" so that HCM will know that these are positions at which connections can be broken.

I/O Operations graphical user interface in HCM

HCM has been enhanced to allow interaction with I/O Operations via HCD for viewing active hardware and ESCON director information. The following new HCM functions have been introduced:

- Graphical view of the active sysplex and its status.
- Manipulation of director matrices.
- Activation of a changed director matrix.
- Issuing I/O Operations commands and viewing the result.

Save list in Edit & Locate windows

The **Locate** object and **Edit** object windows contain a new **Save As...** button. This allows all the listed objects, including the displayed columns, to be saved to a file in a format, that can, for example, be used by a spreadsheet program (such as Lotus® 1-2-3®).

Color printing

HCM now provides the capability of printing hard copies of diagrams in the same colors that appear on the screen.

Generation of a director matrix

HCM provides a new checkbox **Generate default matrix** in the **Create Configuration for Director nn** dialog box that will engage HCD to create a director matrix with appropriate "allows" and "prohibits" based on the IODF information.

Multiple delete

HCM has always permitted selection of several objects for "cropping", but until now it has only supported deletion of a single selected object. It is now possible to delete (disconnect) all the objects selected by pressing the DEL key (or by selecting the **Delete** option from the **Edit** menu).

Other changes

Other improvements are as follows:

- **Comment field in cable assignments dialog.**
  HCM now provides a 32-character editable comment field in the "Assign Cable" dialog. It is visible in the "Cable Assignments" and "Locate Cable" dialog windows.
- **Deeper zooming.**
  HCM has improved the readability of the diagram text on the screen.
- **IODF information in report.**
  When printing a hardcopy report, HCM now includes descriptive information about the associated IODF.
- **Unused cable report.**
  HCM now provides the capability of including unused cables in the hardcopy reports.
- **Selective cabinet reports.**
  Cabinet reports can be limited to selected cabinets.
- **Popup menus.**
  Context-sensitive popup menus have now been provided. They can be accessed with the right mouse key.
- **Refresh background.**
  This option is no longer provided. Once the last dialog has been closed, the diagram will always be completely repainted and hence up-to-date.
- **Disconnect of mutable connections**
  When disconnecting an ESCON link with more than one mutable connection, a dialog is now presented allowing you to select the mutable connection to be deleted.

What's new in HCM for OS/390 V2R7

New utilities

HCM offers you new utilities that make complex and time-consuming tasks easier.

Moving ports

This utility allows moving CHPID-to-control-unit connections that have a path through an ESCON director from one director to another, or rearranging such connections on a single director. The advantage is that you no longer have to manually disconnect and connect the individual objects involved.

Aggregating CHPIDs

This utility allows you to combine attachments of several CHPIDs onto a single CHPID. That is, all control units connected to the source CHPIDs will be disconnected from the source CHPIDs and connected to a target CHPID. That is possible for all CHPIDs connected to the same director.

Migrating CNC to FICON channels

This utility supports you in making the necessary I/O configuration changes, if a new FICON bridge card is installed on an ESCON director. HCM especially supports you if CHPID-to-control-unit connections have to be moved from one port to another port, or if CHPID-to-control-unit definitions have to be aggregated from several CHPIDs to a new FICON CHPID.

New operating system support

HCM now runs as a native 32 bit application under Windows NT® Version 4.0 and Windows 95 or higher. This provides the following benefits:

- Long file names are supported
- Multitasking behavior: Other applications can run while HCM is performing long operations

You can continue to use your current config files.

Network protocol enhancements

Besides APPC, you can now also use TCP/IP as a network protocol to communicate with the HCD server on the OS/390 host.

The APPC communication component has been improved. To use APPC communication you have to provide an updated TP profile.

You do not need data set SYS1.SCBDCOMM on the host any longer.

Navigation improvements

HCM offers you improvements while viewing and navigating through your configuration diagram.

Consolidated View menu	The action **Filter Diagram...** under the **View** menu has been enhanced. The now-called dialog **Change View Attributes** lets you specify all aspects of a configuration's visibility in one single dialog.
Named Views	Using the HCM filter techniques you can tailor a view in such a way that only objects of interest are visible. You can now save such a view for later

reuse. It is possible to save more that one view, thus you can have a set of named views that show different aspects of a configuration.

Enhanced Filter Capability   When choosing **Edit...** and **Locate...** actions from the Menu bar, you now get a dialog that allows you to sort the listed objects according to their attributes and to reduce the list using a filter.

Visualizing ESCON director matrix information

HCM can now visualize information from ESCON director or crossbar switch configurations. The diagram now shows dedicated connections or blocked ports. In addition, when highlighting objects in a diagram, HCM can now use the director configuration data to show what particular route is used for CHPID-to-control-unit definitions. This is especially important when there are definitions in the configuration that have more than one possible physical route from the CHPID to the control unit, for example, when chained directors are involved.

Enhancements to editing director configurations

The presentation of director configurations as a matrix has been significantly improved. The presentation now makes effective use of information of the whole HCM configuration and shows, for example, whether a connection between ports is logically defined by the configuration or not, or whether a connection requires a dedicated connection or might have a dynamic connection.

Furthermore, the enhancements allow you to filter and rearrange ports within the matrix according to your needs.

Tivoli Inventory support

HCM provides support for storing configuration data of S/390 processor configurations in the Tivoli Inventory data base.

Creating cabinets with variable sized panels

You can now create cabinets to contain patch panels with variable individual size. This allows you to have a mixture of panels with different sizes within one cabinet.

Chapter 1. Overview

Traditionally, hardware configuration is carried out on the z/OS or z/VM host using Hardware Configuration Definition (HCD). With HCD you can define the hardware configuration for both the channel subsystem and the operating system. The configuration you define with HCD can consist of multiple processors, each containing multiple partitions. HCD stores the entire configuration data in a central repository, the Input/Output definition file (IODF). The IODF as single source for all hardware and software definitions for a multi-processor or multi-partition system is used to build the IOCDS using the IOCP program and to IPL the operating system.

The z/OS and z/VM Hardware Configuration Manager (HCM) is a PWS (programmable workstation) based client/server interface to HCD that combines the logical and physical aspects of hardware configuration management.

In addition to the logical connections, you can also manage the physical aspects of a configuration. For example, you can effectively manage the flexibility offered by the ESCON/FICON infrastructure (cabinet, cabling). All updates to your configuration are done via HCM's intuitive graphical user interface and, most important, due to the client/server relationship with HCD, all changes of the logical I/O configuration are written into the IODF and fully validated and checked for accuracy and completeness by HCD, thus avoiding unplanned system outages due to incorrect definitions. For z/OS only, HCM also allows you to display operational data such as system status information and operate on the switch via an interface to the I/O Operations function of System Automation on the host.

For more information on how HCM and HCD work together, see Chapter 4, "Building your configuration," on page 63.

Physical and logical data in one place

HCM presents an interactive configuration diagram which allows you to maintain not only the logical connectivity data in the IODF, but also the *physical information about a configuration*. The logical information in the IODF represents the operating system and the channel subsystem definitions; the physical information - cabinets, patchports, crossbar switches, cables, locations and so on - adds the infrastructure to the logical data.

Furthermore, the logical and physical information for each object in the configuration *match* because they are created by the same process. When you create an object, you add its logical and physical information at the same time. When you connect, for example, a controller to a processor, the selected control units are logically defined to the selected CHPID through a controller channel interface; the physical connection, including the cable, is displayed visually in the configuration diagram.

Finding and fixing configuration problems is made much easier since physical and logical information cannot diverge. You can, therefore, use HCM to accurately represent your hardware configuration, by supplementing the IODF data.

Easy to use

HCM's intuitive graphical user interface allows you to navigate easily around the configuration diagram and dialogs to edit information. You can readily create or delete objects, including physical-only objects such as cabinets, patchports, crossbar switches or converters. You can modify any object: editing is as simple as double-clicking on the desired object in the configuration diagram, and adjusting its information in the resulting dialog. After you have created or modified objects, you can physically connect and logically define them. HCM will intelligently adjust the positions of objects in the diagram to clearly display all connections. You can also generate reports and wall charts to analyze, plan, and implement changes to your configuration.

You can tailor your view of the diagram to display only the objects of interest. Zoom in to focus on one area of the diagram, or crop, reorder, show or hide selected objects. HCM will highlight all connections to selected objects.

Although you can perform nearly all configuration tasks in HCM, some functions such as activating a configuration must be done in HCD (see Chapter 14, "HCD-only tasks," on page 313).

Accurate system configuration

The data you define with HCM is used to activate your system. You can use the same data to automatically generate reports and diagrams, eliminating inconsistencies and ensuring accurate documentation of system definitions. By skipping the tedious process of manual data entry, you make fewer errors and save significant amounts of time.

Support for different user groups

HCM addresses the needs of several different user groups involved in configuration management. *System programmers* can use HCM to more easily create and edit configuration definitions that have been traditionally maintained in HCD. For example, HCM supports control unit and device definitions by proposing default parameters.
Hardware support groups can use HCM to add physical data to the configuration; for example, cabinets and patchports which are not dealt with in the IODF.
System planners can generate reports and diagrams to develop future and/or alternate configurations for planning purposes.
Operators can conveniently retrieve information about the active status or about objects in the configuration of a system, and can perform operation tasks on the switch. It is thus possible for the operators to have immediate access to an overview of the configuration to help them analyze and solve problems (this is also possible when HCM is running in stand-alone mode).

Chapter 2. Installing HCM

The installation of HCM is performed in three steps:

1. **Install HCM on the host (z/OS or z/VM)**

   This procedure is performed by the system administrator at the host and is described in the *z/OS Program Directory* for z/OS, or, for z/VM, in the *HCD and HCM for z/VM Program Directory*.

2. **Install HCM on the workstation**

   This process is described for both platforms (z/OS and z/VM) in "Installing HCM on your workstation" on page 5. Before you start this process, ensure that you meet the requirements as stated in "Prerequisites."

3. **Set up the TCP/IP or APPC definitions**

   For z/OS, HCM allows you to use either TCP/IP or APPC as a connection between the workstation and the host. For z/VM, only TCP/IP is possible.

   - If you decide to use TCP/IP, then, depending on the operating system, follow the steps described in "Alternative 1 for z/OS: Setting up TCP/IP definitions" on page 7 or "Setting up TCP/IP definitions for z/VM" on page 25.
   - If you decide to use APPC for z/OS, then follow the steps described in "Alternative 2 for z/OS: Setting up APPC definitions" on page 12.

Furthermore, the following additional information is contained in this chapter:
- "Installing service levels" on page 6
- "Installing HCM in a LAN environment" on page 6
- "Migrating from an earlier HCM release" on page 37
- "Uninstalling HCM" on page 38

Prerequisites

The following requirements must be met to run HCM.

Workstation hardware requirements

HCM runs effectively on a PWS (programmable workstation) with at least a Pentium® 133 MHz and 32 Mbytes of RAM. It is strongly recommended that you have sufficient additional free RAM space for processing the configuration files.

Further requirements are:
- About 50 Mbytes of free disk space
- Color display (1024 x 768 resolution is required)
- VGA display driver
- Communication adapter card

Software requirements

The following software requirements must be met to run HCM.

Workstation

On your workstation (PWS) you need the following:

Operating system: You need to run one of the following operating systems:

- Windows NT Version 4.0 or later
- Windows 2000
- Windows XP

Host communication: Either TCP/IP (for z/OS or z/VM) or APPC (possible for z/OS only) is required.

- TCP/IP communication: TCP/IP networking protocol must be installed (delivered with Windows XP, Windows 2000, Windows 98 or Windows NT). For z/OS, to use HCM with TCP/IP connection, UNIX® System Services and TCP/IP must be running and the HCM user ID must be set up to entitle it to access the UNIX System Services. **Note:** Superuser authority is not necessary, but a home directory is to be provided for the user.
- APPC communication: IBM Personal Communication AS/400® and 3270 for Windows XP, Windows 2000, Windows 98, Windows NT (Version 4.11 or later) is required.

HCM installation: You need a method to download the code from the host to the workstation (for example, FTP or PCOMM).

Tivoli Inventory support: If you want to use the Tivoli Inventory support (optional), you need the following:

- The HCM workstation must be made a PC Managed Node of a Tivoli Management Region (TMR). The appropriate PC Agent and the Tivoli Inventory PC Scanning Program must be installed to make the HCM workstation a PC Managed Node.
- HCM supports version 3.2 of Tivoli Framework and Tivoli Inventory. The following patches must be installed:
  - 3.2-TMP-0023
  - 3.2-INV-0001

Host

HCM is a client/server application using HCD on the host as its server. The communication protocol between the workstation and the host is either TCP/IP or APPC (only possible for z/OS). Although HCM 2.9.0 is shipped with z/OS Version 1 (as well as with z/VM Version 4.4), it can communicate with an HCD of OS/390 1.3 or later. HCM detects whether all HCM functions can be run with a backlevel HCD or whether PTFs need to be installed.

The following table shows the PTFs that need to be installed for various z/OS operating system releases.

Table 2. PTFs

Release	PTF
OS/390 2.9 OS/390 2.10 z/OS 1.1 z/OS 1.2 z/OS 1.3 z/OS 1.4 z/OS 1.5 z/OS 1.6	UW65627, UW75342, UW74689, UW73140, UW71634, UW99307, UW99308, UW69960, UW68998, UW68962, UW68656, UW99344, UW99345, UW99293, UW99294, UW67839, UW67844, UW76423, UW76445, UW76450, UW90706, UW90707, UW99394, UW99395, UW77051, UW80358, UW99413, UW99414, UW99420, UW99421, UW99422, UW92937, UW92938, UW83726, UW83735, UW85018, UW85019, UW87255, UW87516, UW87517, UW89117, UW89118, UW93969, UW93970, UW90746, UW90747, UW94407, UW94408, UW90748, UW90749, UA01328, UA01069, UA02801, UA02802, UA02016, UA02426, UA02427, UA02428, UA02429, UA04685, UA04686, UA03355, UA03354, UA03799, UA03800, UA03755, UA03765, UA90070, UA90071, UA90072, UA05456, UA06641, UA06700, UA06843, UA06846, UA07314, UA08089, UA08179
z/VM 4.4 z/VM 5.1	UM31044, UM30896, UM30869, UM30754

I/O Operations functions: To be able to use the *I/O Operations* functions offered in HCM, (see Chapter 11, "User interface to I/O Operations," on page 221) the ESCON Manager Version 1 Release 3 (including PTFs UN94223 and UN94761) or later, or System Automation for OS/390 (I/O Operations) must be installed and running.

Note: Access to the *I/O Operations* tasks is not possible with HCD versions earlier than OS/390 Version 2 Release 9 HCD. Also, access to *I/O Operations* is not possible using z/VM HCD.

Installing HCM on your workstation

> **Note:**
> The information contained in this section is valid for operating systems z/OS and z/VM.

Before you can install HCM on your workstation, the code needs to be installed on the z/OS host using SMP/E or on the z/VM host using VMSES/E. You can then start to download the HCM installation utility on your workstation. Proceed as follows:

1. The HCM installation utility is available
   * in the data set member SYS1.SEEQINST(EEQINSTX) on the z/OS host
   * or in file EEQINSTX EXEBIN on the z/VM host from minidisk 4VMHCD40 400.

   Download the installation utility in *binary* mode onto your workstation into a temporary directory. Rename the file such that it is executable (that is, it must have the extension EXE). To install HCM, run the file on your workstation.

> **Note:** If the product has been installed on your host system in a different data set, ask your system administrator for its name and use this name for the download.

2. To install HCM, invoke the HCM installation utility that now resides on your workstation. For example, choose the **Run** option in the **Start** menu. The installation starts.

3. Follow the steps through the installation. During the installation you are asked for the destination location of HCM and the configuration files. You can choose a new folder or select an existing one by clicking the **Browse** button.

4. Once the installation has successfully completed, HCM is installed on your system and you can delete the HCM installation utility. It is not necessary to reboot your system. The HCM installation creates an entry in your **Start** menu from where you can launch HCM.

Installing service levels

From time to time IBM makes available service levels to update HCM. Each service level contains the complete product and replaces prior HCM service levels.

During installation of the service level, the installation utility informs you that HCM is already installed on your system. It is recommended to uninstall the old service level before installing the new one.

Always see the EEQREAD.ME file before using the new service level. It resides in the product directory after you have installed the new service level.

Installing HCM in a LAN environment

For shared access of HCM you can install HCM in a LAN environment. That has several advantages.

When you keep the HCM product directory on a LAN disk, all HCM users are using the same service level of HCM and a service update has only to be installed once.

Another possibility is to keep the HCM configuration files on a common LAN resource. This has the advantage, that the configuration data is accessible for different HCM users without the necessity of copying or transferring files. Remember the usual rules for multiple access of files in this content. Make sure that only one user has write access to the files at one time. HCM configuration files contain important data, therefore it is important to ensure, that these data do not get lost or damaged. In some cases it is easier to backup files from a LAN disk than to backup files from individual workstations.

When you run HCM in a LAN environment, consider the following:

- The installation utility of HCM creates an entry in the **Start** menu of the PWS from which HCM has been installed. If you want to use HCM from the LAN resource, you must create an icon on your desktop to work with HCM.

- In case of a fatal error, or if some logging mechanism is activated for debugging purposes, HCM writes some log files to the product directory. If in your installation the HCM product directory on the LAN is write protected, the log files could not be created. In addition, if you want each LAN user to keep own log files, you can specify a directory other than the default one, using the ERRLOGDIR parameter in the [HCM] section of the EEQHCM.INI file. See "The HCMERR.LOG PWS file" on page 390 for more information on the error log file.

- As the installation utility keeps some data locally on the workstation from which HCM is installed, it is recommended that the HCM program on the LAN is only deleted and updated by the PWS from which it has been installed originally.

Establishing the host communication

After you have successfully installed HCM on your workstation, you have to make the definitions to link the host and the workstation. You can select from the following methods:

- "Alternative 1 for z/OS: Setting up TCP/IP definitions"
- "Alternative 2 for z/OS: Setting up APPC definitions" on page 12
- "Setting up TCP/IP definitions for z/VM" on page 25

Alternative 1 for z/OS: Setting up TCP/IP definitions

When the HCM client is running on your workstation, it needs a server program running on the host. The server program is a TCP/IP program that listens for incoming HCM requests on a specific TCP/IP port. These HCM requests are passed to HCD to be executed. The server program - let it be named HCM agent - has to be started before HCM requests are passed. The HCM agent (HCD server) is started by a daemon program (HCM dispatcher) as soon as a HCM login request has been issued. This HCM dispatcher program must be started before the first HCM login request is issued. It can run permanently on the z/OS host.

The following picture provides an overview of the structure and illustrates the relations between the HCM client, HCM dispatcher, and the HCM agent.
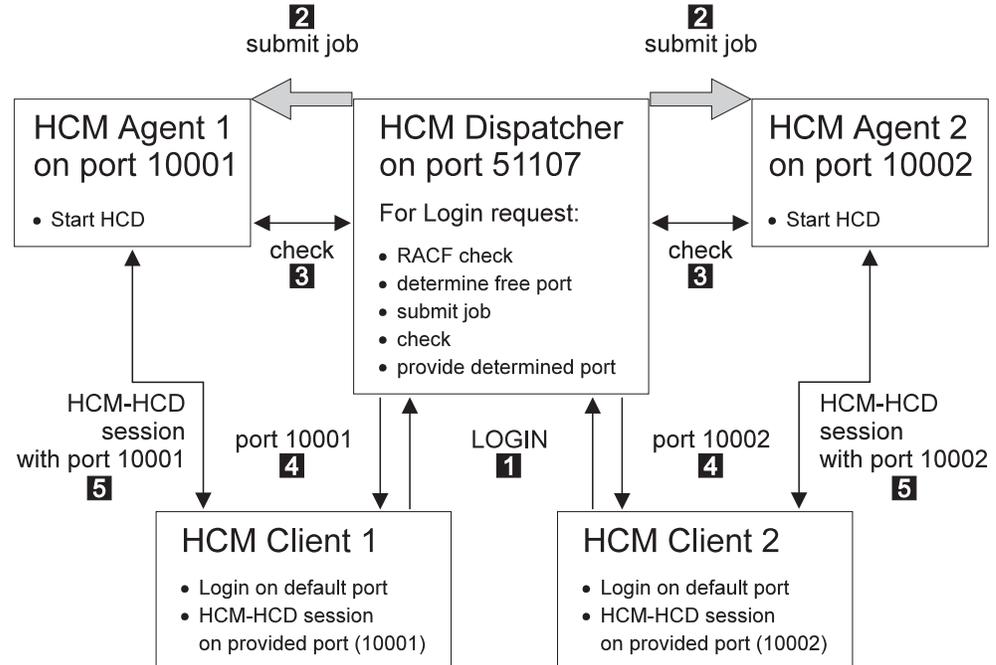
Figure 1. Relationship between HCM Client, HCM Dispatcher, and HCM Agent

The HCM dispatcher listens on a specific TCP/IP port and waits for incoming HCM login requests. For each HCM login request **1**, the HCM dispatcher checks the passed user ID and password for correctness. If user ID and password are correct, the HCM dispatcher looks for a free IP port. Then it starts an HCM agent

program (HCD server program) which listens to HCM requests on that particular IP port **2** . As soon as the HCM agent is started and ready **3** , the HCM dispatcher passes the particular IP port to the HCM client **4** . The HCM client then closes the session to the HCM dispatcher and starts a session to the started HCM agent using the passed IP port **5** . When the HCM client has connected the HCM agent, the HCM dispatcher is free again to wait for other incoming HCM login requests on its IP port. As soon as HCM terminates the HCM client server connection, the HCM agent will be terminated and the used port is freed again.

The advantage of having an HCM dispatcher which waits permanently on a specific port for incoming HCM login requests is, that each HCM user performs a login request to a fixed port ID and does not have to specify a particular job input to start the HCD server. This means, that all HCM users automatically have the same setup, and that this has only to be done once. As the HCM dispatcher is always running, all HCM users always use the same IP port for the login request. HCM saves the IP port across sessions and offers it in the login window. Therefore, the user has to specify it only once.

The TCP/IP port for the HCM login requests is determined when the HCM dispatcher is started. If during the start of the HCM dispatcher nothing special is specified, the default TCP/IP port number is 51107. This is also the default port number which is used by HCM for a login request if no port is specified.

The HCM dispatcher creates a job out of a skeleton and submits this job to start the HCM agent (for an example of the job skeleton, see "Skeleton used to start the HCM agent" on page 9). After the HCM agent has been started and is running, the HCM client communicates with the HCM agent. The HCM client uses the same host name for the communication with the HCM agent as it has used for the login request to the HCM dispatcher. Therefore, the HCM agent must run on a system with the same host name as the HCM dispatcher. This fact might be especially important, if your system is within a parallel sysplex. In this case, you can specify the system on which the agent must run in the provided skeleton.

You can start the HCM dispatcher in two ways:
- Starting the HCM dispatcher as a started task by using the procedure CBDQDISP provided in the library SYS1.PROCLIB. Consider to start the HCM dispatcher automatically after IPL of your z/OS system (for example, by using System Automation). See "Starting the HCM Dispatcher as a started task" on page 9 for an example on how to start the HCM dispatcher as a started task.
- Starting the HCM dispatcher by submitting a batch job. A sample of a job, which can be used to start the HCM dispatcher is provided as CBDQDISJ in 'SYS1.SAMPLIB'. Adapt this job, before you submit it.

If you do not want to accept the default port number, you can choose your own by changing the procedure or the sample job to start the HCM dispatcher. Inform the HCM users of the TCP/IP port for the login requests. The HCM client will not accept a port number 0 or port numbers bigger than 65535.

Each HCM session needs its own dedicated server, and each active HCM server needs its own unique TCP/IP port number. The HCM dispatcher looks for a free port number in a specific range. Per default a port number is chosen in the range of 10000 to 65535 for the server. It is possible to determine a different range for the ports to be chosen for the HCM agent during start of the HCM dispatcher.

The user ID under which the HCM dispatcher is running, as well as the user IDs used for working with HCM must have permission to use UNIX System Services. Note that superuser authority is not required, but a home directory is to be provided for these user IDs. For example, you can use /u/*userid* (where *userid* is your own user ID) as a home directory for the HCM user ID.

Depending on your general TCP/IP setup, it might be necessary that you include a DD statement for SYSTCPD for the HCM dispatcher and the HCM agent.

Skeleton used to start the HCM agent

There is a sample of skeleton CBDQAJSK provided in SYS1.PROCLIB. The HCM dispatcher uses this sample job to build up a job, which is submitted to start the HCM agent.

You can adapt this skeleton according to your installation needs. Especially you may want to specify accounting information in the job card of the skeleton, if your installation requires accounting information. Note, if your installation requires accounting information, and there is no accounting information specified in the skeleton, each HCM user will have to provide this information in the EEQHCM.INI file. For all other substitutable parameters, the HCM dispatcher provides default values, for example, &SYSUID..HCD.PROFILE and &SYSUID..HCM.TRACE as data set names for the HCD profile and the HCM trace data set. If you specify values for substitutable parameters, then these values will be used and not the default values or values specified in the HCM user's EEQHCM.INI file, for example, an increased REGION size for the HCM agent.

In any case, check, whether you have to adapt this skeleton for your environment regarding the JOBPARM SYSAFF parameter for JES2 or MAIN SYSTEM parameter for JES3. The job must be executed on the system with the system name specified during HCM client logging to the HCM dispatcher.

Furthermore, an HCM user can take this skeleton, make a copy of it, and specify values for personal needs. To use this private copy, the HCM user needs an entry in the EEQHCM.INI file to tell the HCM dispatcher not to use the default skeleton, but the user specified skeleton.

Starting the HCM Dispatcher as a started task

There is a procedure CBDQDISP provided in SYS1.PROCLIB, which you can use to start the HCM dispatcher as a started task.

You can create a new user ID or use an existing one to be associated with the task of the HCM dispatcher. This user ID has to have permission to use UNIX System Services.

After the procedure has been adapted to your installation needs, you can start it by using the start command. Start the HCM dispatcher always after the system has been IPLed. You can also start the HCM dispatcher automatically using System Automation for OS/390.

You can change the following parameters in the procedure:

Job name
> The HCM dispatcher submits a job for each incoming HCM login request. Per default the job name of this job starts with CBD. If you want the job names to start with something different than CBD, you can specify a different string for the beginning of the job name by setting the JNP variable. If you want the job name to contain the user ID of the HCM user,

specify +U and the HCM dispatcher will substitute the +U with the requester's user ID. You can also add a prefix to the user ID or append a string to the user ID. For example, a X+UY for a passed user ID BMGN would result in XBMGNY for the beginning of the job name. Note, the HCM dispatcher does not accept more than 7 digits for the beginning of a job name. Strings longer than 7 digits are truncated. The HCM dispatcher generates a job name using the JNP variable and fills it up to 8 digits. If the JNP variable contains 3 digits, the job name is filled up to 8 digits by using parts of the port address of the HCM dispatcher and parts of the port address of the HCM client, which will be started. If the JNP variable does not have 3 digits, it will be filled up to 8 digits by using 0...9 and A...Z.

Logging information
For special cases (debugging or better control), the HCM dispatcher can write logging information into a data set. In this case, change the LOG variable.

Port If there is any reason to use another port than 51107 on which the HCM dispatcher listens for incoming HCM login requests, you can specify your port by setting the PORT variable.

Skeleton
You can specify another skeleton to be used to start the HCM agent by changing the JSK variable.

Port range
You can use a different range of ports, in which the HCM dispatcher will determine a free port to be used by the HCM agent. Set the variables P0 and P1 to appropriate values. Note, ports bigger than 65535 are not allowed.

Creating a user ID: If you want to run the HCM dispatcher as a started task, you must create a user ID for it.

1. Create a user ID to be used as started task for the procedure CBDQDISP. This user ID must have permission for running UNIX System Services.
2. Define the user ID to be used for the started task procedure CBDQDISP.
3. Refresh RACF®.

The following figure shows a sample job in which the user ID for the HCM dispatcher can also be used by the UNIX System Services. (Note that the specified home directory for the HCM dispatcher is the root directory in this example).

```
//ADDUSER  EXEC PGM=IKJEFT01
...
//SYSIN    DD *
   AU CBDQDISP NAME('STARTED-T. M. GNIRSS') OWNER(STCGROUP) +
      DFLTGRP(STCGROUP) +
      OMVS(HOME(/) PROGRAM(/bin/sh) UID(4711))
//*
//DEFRACF  EXEC PGM=IKJEFT01
...
//SYSTSIN  DD *
   RDEF STARTED CBDQDISP.*  STDATA(USER(CBDQDISP)  GROUP(STCGROUP))
//*
//REFRESH  EXEC PGM=IKJEFT01
...
//SYSTSIN  DD *
   SETR REFRESH RACLIST(STARTED) GENCMD(*) GENERIC(*)
/*
```

Figure 2. Sample JCL for Creating a User ID for UNIX System Services

Starting the HCM Dispatcher as a batch job

You can either start the HCM dispatcher as a started task, or by submitting a batch
job. For this purpose, you can use the sample job CBDQDISJ in SYS1.SAMPLIB and
submit it after you have adapted it to your needs.

Stopping the HCM Dispatcher

To stop the HCM dispatcher, use the **cancel** command. The **stop** command is not
supported.

Define an HCD profile

HCM uses HCD on the host as its server. This allows you to define an HCD profile
to tailor HCD supplied defaults and processing options to your installation needs.
Using a profile is optional. The profile file data set must have the following
characteristics:
* Be either a sequential or a member of a partitioned data set
* Have fixed length (80 bytes), fixed blocked record format

For more information, see *z/OS HCD User's Guide*.

Note: If you do not need an HCD profile, remove the substitutable variable
        <HCDPROF> from the skeleton, which is used to start the HCM agent.

Allocate the HCM trace data set

Before you can start an HCM agent (HCD server) on the host, you have to allocate
a data set that will be used by this HCM agent's trace facility. This trace data set
must have a different name than the standard HCD trace data set (which is called
userid.HCD.TRACE), allowing you to use HCM and HCD concurrently. The
recommended name for the HCM agent's trace data set is *userid*.HCM.TRACE.

The following job can be used to allocate the trace data set.

```
//ALLOC    JOB (DE03141,,),'GNIRSS',CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//*
//STEP1    EXEC PGM=IEFBR14
//DUMMY    DD DSN=WAS.HCM.TRACE,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//         SPACE=(CYL,(5,20)),DISP=(NEW,CATLG),UNIT=SYSALLDA
//
```

Figure 3. Sample Job for Trace Data Set Allocation

Specifying TCP/IP on your workstation

Before you can use HCM with TCP/IP, the TCP/IP networking protocol must be installed and configured on your workstation. See your TCP/IP documentation for information on how to do that.

Furthermore, see the HCM User's Guide update for service level 7 on the HCM home page for an example on how to configure TCP/IP on your workstation. See **http://www.ibm.com/servers/eserver/zseries/zos/hcm**, click on **Product Updates**, select the **HCM Documentation**, and view the User's Guide updates for **OS/390 2.9.0 HCM (TCP/IP Support)**. The update is available in PDF format. You can view it with the Adobe Acrobat Reader.

Alternative 2 for z/OS: Setting up APPC definitions

This chapter describes how to make your APPC definitions on the host and your workstation.

Setting up APPC requires updates at each target host, at the controller being used for the communication, and at the workstation on which the workstation feature is running. These tasks and how to verify a setup are described in the following sections by means of the example shown below. (The workstation is configured as an independent LU.)
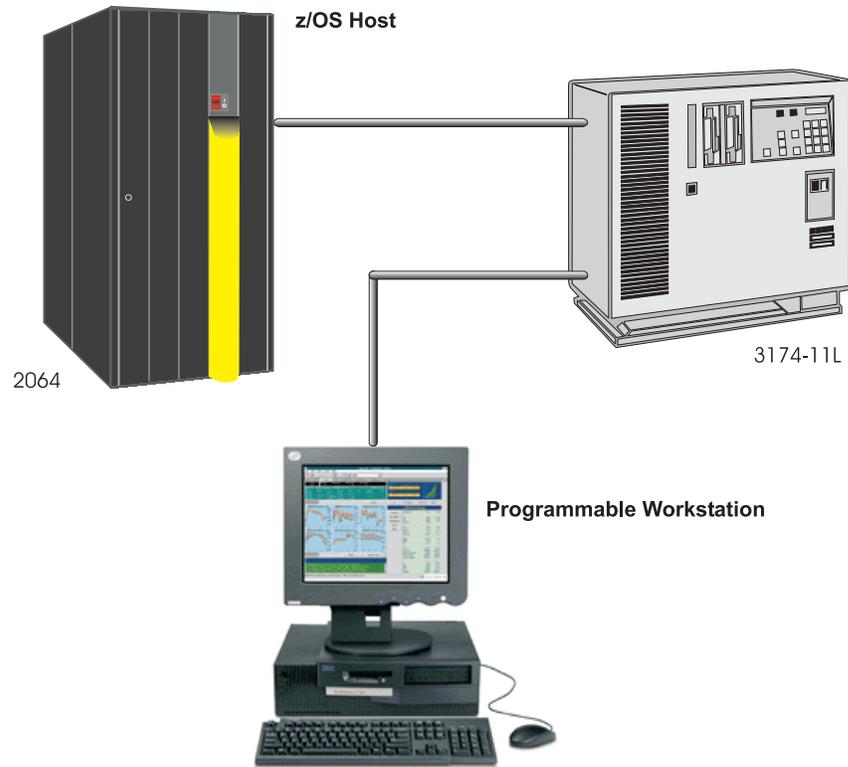
Figure 4. Example of a Setup with a z/OS Host, a PWS, and a Controller

APPC setup values you need to enter and where

The data values used in the example are listed sequentially below. These values are identified by digits and letters. The letters indicate where to enter the value: at the **H** = host, **C** = controller, and **W** = workstation.

Where	Value	Description
`1H`	SYS1.APPCTP	The VSAM file containing the TP profiles that will be referenced by APPC to start the host transaction program (TP).
`2HW`	EEQTPA00	The name of the transaction program. This value is sent from the workstation to the host, at the time a conversation is allocated, to tell APPC which TP profile entry is to be used to initialize the host transaction program.
`3H`	HCM	The class of transaction initiator that will run the host TP to define job scheduling and job performance characteristics. (The name you choose must also be defined in SYS1.PARMLIB(ASCHPM*xx*)
`4H`	IPPAPPC	The VTAM® mode table that contains - in our example - #INTER.
`5W`	MVSSYSTM	The symbolic destination name (SDN) given to the host system. This name must be specified on the **Login** window in order to setup a communication link with the host. This value relates the host name known to you to a partner LU, Mode Name (VTAM), and TP Name (APPC). You must use capital letters for the SDN.
`6HW`	IPSA7APC	An ACB name of the host that is used as the partner LU to establish the session with the workstation. (You can also define an alias.)

Where	Value	Description
7HW	#INTER	The mode name, or logmode, that is an entry in the VTAM mode table, which is listed as Number 1. A network node server, as in this example, will route APPC requests to the target system; therefore, connections and partner LU location definitions need not be performed.
8HW	FDJ81006	Both the control point (CP) name of this workstation and the name of the independent LU that will establish a session with the host. (You can also define an alias.)
9HW	DEIBMIPS	The ID of this network. You can find the NETID in the VTAM start options list. Look in your file ATCSTR*xx*, where *xx* is installation-dependent.
10CW	4000 1000 0001 04	The LAN destination address of the token-ring adapter in the network's communication controller, including the 2-character service access point (SAP) ID of 4.

APPC enablement on the host

You need to make APPC definition on your host. See the following publications for information about creating an APPC environment in your installation.

- *z/OS MVS Planning: APPC/MVS Management*, SA22-7599
- *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621

Step 1: Define TP profile: The first step in the process of setting up the APPC communication is to define an APPC TP profile. This TP profile contains JCL statements that call a job to initialize HCM. ATBSDFMU is the APPC administration utility program.

```
//userjob  JOB (account),'info',MSGCLASS=A,MSGLEVEL=(1,1),
//             NOTIFY=user,REGION=7000K,CLASS=A
//*************************************************************
//*  This job adds the TP profile EEQTPA00 used by HCM 2.7.0  *
//*  to the profile data set SYS1.APPCTP.                     *
//*************************************************************
//*     LICENSED MATERIALS - PROPERTY OF IBM                  *
//*     5647 - A01                                            *
//*     (C) COPYRIGHT IBM CORP. 1999                          *
//*     STATUS = XXXXXXX                                      *
//*************************************************************
//STEP     EXEC PGM=ATBSDFMU
//SYSPRINT DD   SYSOUT=*
//SYSSDLIB DD   DSN=SYS1.APPCTP,DISP=SHR                  1H
//SYSSDOUT DD   SYSOUT=*
//SYSIN    DD   DATA,DLM=XX
    TPADD
        TPNAME(EEQTPA00)                                 2HW
        SYSTEM
        ACTIVE(YES)
        TPSCHED_DELIMITER(##)
          TAILOR_SYSOUT(NO)
          TAILOR_ACCOUNT(NO)
          CLASS(HCM)                                     3H
          TPSCHED_TYPE(STANDARD)
          JCL_DELIMITER(END_OF_JCL)
//EEQTPA00 JOB  MSGLEVEL=(1,1),MSGCLASS=0,REGION=128M      see note 1
//HCMA00    EXEC PGM=CBDQCSRV,TIME=NOLIMIT
//STEPLIB  DD DISP=SHR,DSN=SYS1.SCBDHENU                   see note 2
//CEEDUMP  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//HCDPROF  DD DISP=SHR,DSN=&SYSUID..HCD.PROFILE           see note 3
//HCDTRACE DD DISP=SHR,DSN=&SYSUID..HCM.TRACE             see note 4
END_OF_JCL
  KEEP_MESSAGE_LOG(ERROR)
  DATASET_STATUS(NEW)
  MESSAGE_DATA_SET(&SYSUID.&SYSWUID.&TPDATE.&TPTIME.JOBLOG)
##
XX
/*
```

Figure 5. APPC Transaction Profile - EEQTPA00

This sample job is shipped with the product in SYS1.SEEQINST member
EEQTPA00.

Notes:

1. The value used for the region depends on your needs. If you use a value that is
   too small, you can run out of storage (HCD messages CBDA056I or CBDA474I
   will be issued to inform about this condition).
2. SYS1.SCBDHENU contains HCD messages which are also displayed in HCM.
   However, if you have decided for your installation that this library was moved
   into another library (for a faster access of the messages), you have to adapt the
   STEPLIB to your needs.
3. This is the HCD profile. See "Step 4: Define HCD profile" on page 16 for more
   details.
4. This is the trace data set needed when tracing HCD activities. See "Step 5:
   Allocate the HCD trace data set" on page 16 for more details. Note that the
   data set needs to be in read/write access mode.

Step 2: Define in SYS1.PARMLIB(APPCPMxx): In the second step you define the host LU to which the workstation (HCM) establishes a session. In addition, you define the data set where you can find the TP profile used to establish a communication.

Sample output is shown below for our example—SYS1.PARMLIB(APPCPM00).

```
SYS1.PARMLIB(APPCPM00) - DEFINES TO MVS/APPC, THE LOCAL LU THAT
                         HCM WILL TALK TO.

LUADD ACBNAME(IPSA7APC),                        6HW
      SCHED(ASCH),
      TPDATA(SYS1.APPCTP),                      1H
      TPLEVEL(USER)
```

Step 3: Define in SYS1.PARMLIB(ASCHPMxx): The next step is to tell the APPC translation scheduler the scheduling characteristics of the transaction program on the host that will communicate with the workstation.

Sample output is shown below for our example—SYS1.PARMLIB(ASCHPM00).

```
SYS1.PARMLIB(ASCHPM00) - DEFINES TO THE APPC/MVS TRANSACTION
                         SCHEDULER, THE CHARACTERISTICS OF THE
                         TRANSACTION PROGRAM ON THE HOST THAT
                         WILL BE IN CONVERSATION WITH HCM.
CLASSADD CLASSNAME(HCM),                        3H
      MAX(10),
      MIN(2),
      RESPGOAL(0.02),
      MSGLIMIT(500)
OPTIONS DEFAULT(HCM),                           3H
      SUBSYS(JES2)
TPDEFAULT REGION(4M),
      TIME(1440),
      MSGLEVEL(1,1),
      OUTCLASS(A)
```

Step 4: Define HCD profile: HCM uses HCD on the host as its server. This allows you to define an HCD profile to tailor HCD supplied defaults and processing options to your installation needs. Using a profile is optional. The profile data set must have the following characteristics:
- Be either a sequential or a member of a partitioned data set
- Have fixed length (80 bytes), fixed blocked record format

For more information, see *z/OS HCD User's Guide*.

Note: If you do not need an HCD profile, remove the HCDPROF DD statement from the EEQTPA00 TP profile.

Step 5: Allocate the HCD trace data set: Before you can start an HCD server on the host, you have to allocate a data set that will be used by the HCD trace facility. This trace data set must have a different name than the standard HCD trace data set, allowing you to use HCM and HCD concurrently. The recommended name is userid.HCM.TRACE.

If you decide to use another name for the trace data set, you must change the EEQTPA00 TP profile accordingly.

The following job can be used to allocate the trace data set.

```
//ALLOC    JOB (DE03141,,),'GNIRSS',CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//*
//STEP1    EXEC PGM=IEFBR14
//DUMMY    DD DSN=WAS.HCM.TRACE,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//          SPACE=(CYL,(5,20)),DISP=(NEW,CATLG),UNIT=SYSALLDA
//
```

Figure 6. Sample Job for Trace Data Set Allocation

Step 6: Make the necessary VTAM definitions: The following VTAM definitions
are required in order to establish an HCM connection to HCM on the host using
an APPC communication link.
1. Specify an application program major node to define the host LU to VTAM.
2. Specify a local SNA major node to define the 3174 and the workstation to
   VTAM.
3. Optionally you may define a mode table with the appropriate log mode entry,
   if you are not using either #BATCH or #INTER. Assemble and link this mode
   table into your load module library.

Define the host LU to VTAM: Define an application major node for the host LU that
will be in session with HCM. The numbers are explained in "APPC setup values
you need to enter and where" on page 13.

```
A7APPC   VBUILD TYPE=APPL              APPL MAJOR NODE FOR APPC
IPSA7APC APPL  ACBNAME=IPSA7APC,               6HW
               APPC=YES,
               AUTOSES=0,
               DDRAINL=NALLOW,
               DRESPL=NALLOW,
               EAS=20,
               MODETAB=IPPAPPC,                4H
               DLOGMOD=#INTER,                 7HW
               PARSESS=YES,
               SECACPT=CONV,
               SRBEXIT=YES,
               VERIFY=NONE,
               VPACING=2
```

Define local SNA major node for controller and workstation: In the example, a 3174
establishment controller is used. The sample output is shown below. The numbers
are explained in "APPC setup values you need to enter and where" on page 13.

```
LOCAL SNA MAJOR NODE FOR THE 3174 & LAN-ATTACHED WORKSTATION

IPPLAN04 VBUILD TYPE=LOCAL
*
LOCAL500 PU    CUADDR=500,PUTYPE=2,MAXBFRU=12,ISTATUS=ACTIVE,
               SSCPFM=USSSCS,VPACING=0,XID=YES
* INDEPENDENT LUS
FDJ81006 LU    LOCADDR=0,DLOGMOD=#INTER,MODETAB=IPPAPPC      8HW  7HW  4H
* DOWNSTREAM PU FOR 3270 SESSIONS
PU504    PU    CUADDR=504,PUTYPE=2,MAXBFRU=12,ISTATUS=ACTIVE,
               DLOGMOD=ESAAPPC,MODETAB=OS2VM,
               PACING=1,SSCPFM=USSSCS,VPACING=1
LOLN5041 LU    LOCADDR=2,DLOGMOD=SNX32702,MODETAB=ISTINCLM,
               USSTAB=USSIPLA
```

Security information: To use HCM, you must have a user ID and password
defined to the host security software (for example RACF).

Furthermore, since HCM automatically accesses SYS1.NUCLEUS you need read access permission for SYS1.NUCLEUS.

Planning APPC enablement on the workstation

HCM uses a LU 6.2 type communication link to HCM on the host. The LU (logical unit) is accessed via the Common Programming Interface - Communications (CPI-C) application interface. This section describes the steps needed to create a communication link.

SNA provides great flexibility in defining the layout of the network. It is possible to configure your workstation as Type 2.1 node, attaching directly to an SNA subarea. It is also possible to incorporate a workstation as an APPN network node. This chapter does not cover all possible configurations, it shows only one example out of several configuration possibilities.

You have to enter some information such as the LU name or TP profile name in different places. Using different values will result in a non-operational link. Before starting your actual configuration, you should collect all information needed at a later stage. Make sure you find proper answers to the following questions before continuing.

- What is the name of the LU to be used by the HCM server side of the host? Ask your SNA network administrator for the name of the LU. It has to be an LU 6.2 type. This will be used in the workstation, APPC and VTAM definition. It will be referred to as the host LU in this chapter.
- What is the name of the network, the host LU belongs to? Remember, that a fully qualified destination name consists of the network name and the LU name.
- Agree on the name of the transaction program (TP) registered with APPC on the host. Ask your system administrator responsible for the host system for a valid name. The proposed name is EEQTPA00.
- What is the name of the LU to be used on the workstation? It is usually the same name as the local node name. Note, Local node names are also referred to as CP or PU name. Unless you use VTAM 4.1 or later, this name must be defined in VTAM as an independent LU (LOGMODE=0). This name will be referred to a local LU name in this chapter.
- What is the mode name to classify the transport characteristics? This name must be a name supported by the mode table which serves your local LU name. Check with your SNA networking specialist, and ask which modes are supported with regard to your local LU.
- What is the Token-Ring address of the boundary node communicating to the SNA subarea network? If you have already defined 3270 sessions via a Token-ring connection, make sure to use the same link for the APPC connections.

Defining APPC on your workstation

After you have installed IBM Personal Communications on your workstation you can start to make your APPC definitions.

1. Select **SNA Node Operations** that you find in the **Start** menu in **Personal Communications** under **Administrative and PD Aids.**
2. In the **SNA Node Operations** dialog, select **SNA Node Configuration** under the **Launch** menu. In this dialog, you will have to select several configuration options.
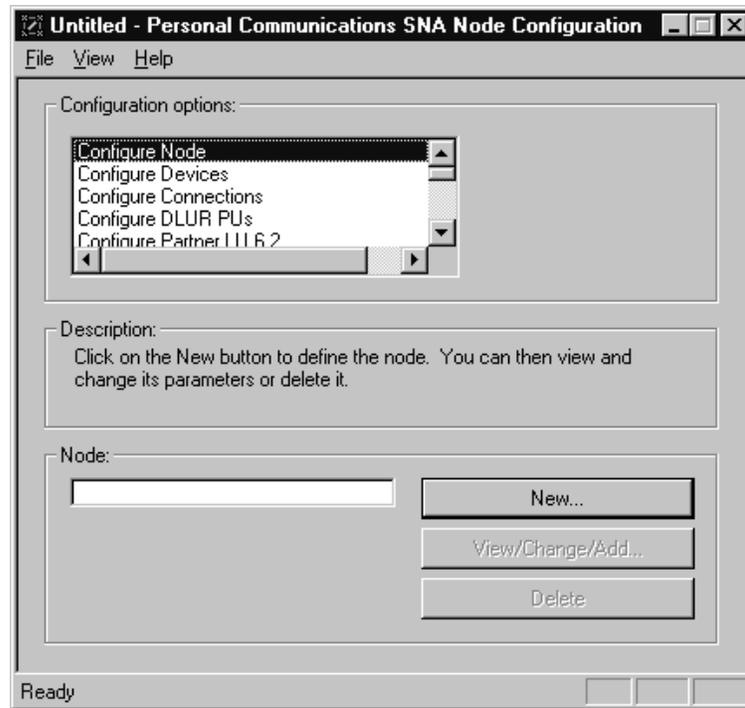
Figure 7. SNA Node Configuration

3. First, select **Configure Node** and press the **New** button.

   a. The **Define the Node** dialog appears. Fill in the **Fully qualified CP name** that consists of the network ID and the LU name of the workstation. You also have to specify a CP alias and the local node ID. Then select **OK**.
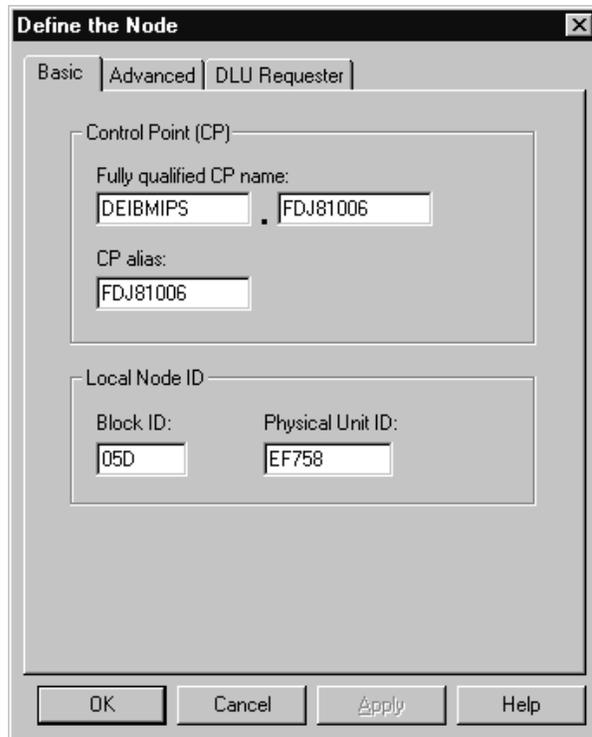
Figure 8. Define the Node

4. Back in the **SNA Node Configuration** dialog, select **Configure Devices**. The DLC type in this sample is LAN. Press the **New** button.

   a. The **Define a LAN device** dialog appears. You can accept the offered values and press **OK**.
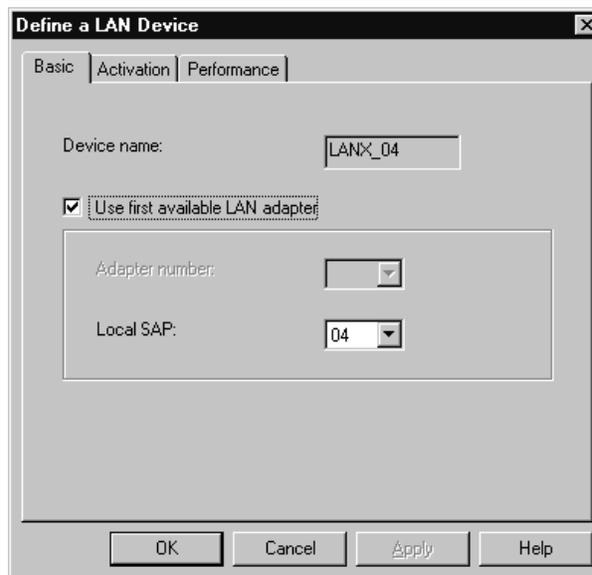
Figure 9. Define a LAN Device

5. Back in the **SNA Node Configuration** dialog, select **Configure Connections** and press the **New** button.

a. The **Define a LAN Connection** dialog appears. Fill in the destination address.
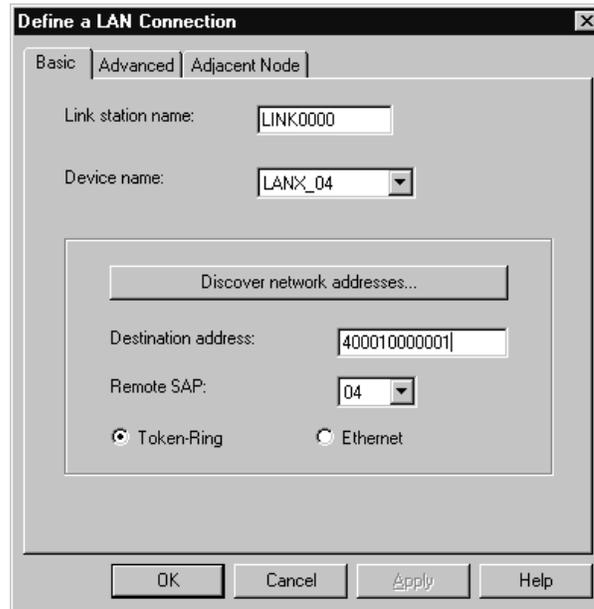
Figure 10. Define a LAN Connection

b. Check the **Advanced** page of this dialog.
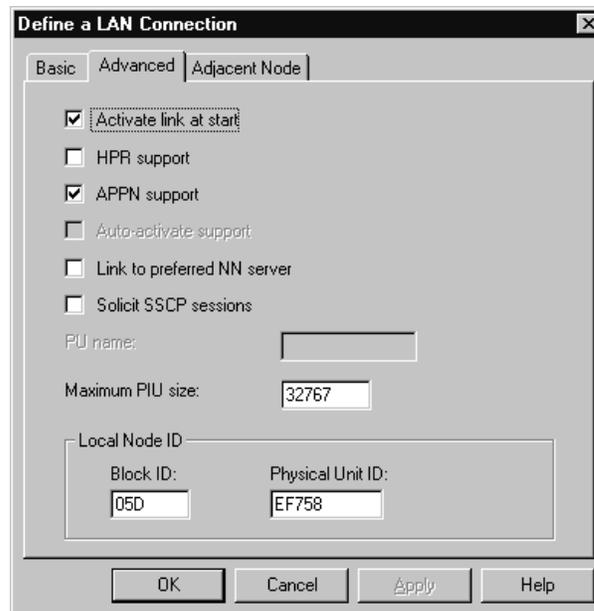
Figure 11. Define a LAN Connection, Advanced

c. Select the **Adjacent Node** page of this dialog. Fill in the **Adjacent CP name** and select your **Adjacent CP type**. Press **OK**.
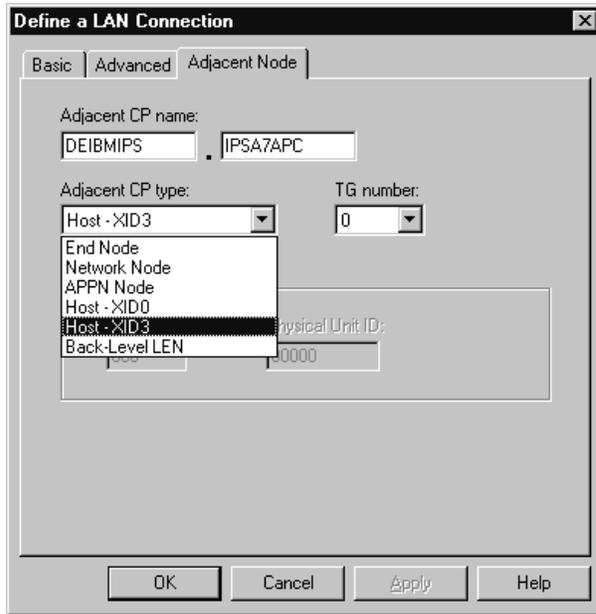
Figure 12. Define a LAN Connection, Adjacent Node

6. Back in the **SNA Node Configuration** dialog, select **Configure Partner LU 6.2** and press the **New** button.

   a. The **Define a Partner LU** dialog appears. Fill in the partner LU name and an alias to which HCM has to establish a connection. Select also the CP name.
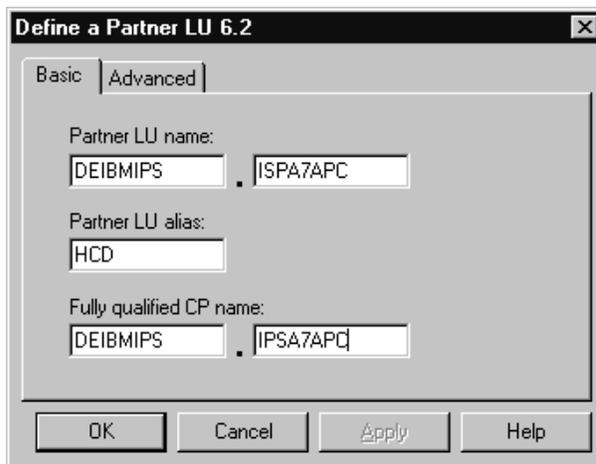
Figure 13. Define a Partner LU

   b. Check the **Advanced** page of this dialog. You can accept the offered values and press **OK**.
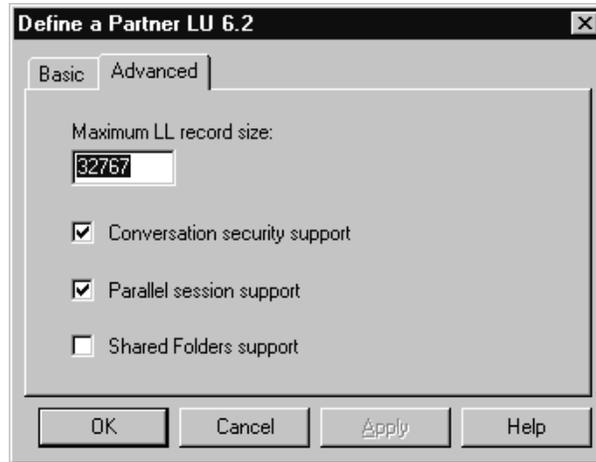
Figure 14. Define a Partner LU, Advanced

7. Back in the **SNA Node Configuration** dialog, select **Configure CPI-C Side Information** and press the **New** button.

   a. The **Define CPI-C Side Information** dialog appears. Specify a **Symbolic destination name** of your choice. You will have to enter this name in the Login dialog of HCM to establish a host connection. By using this name, APPC knows to which host **(Partner LU)** a connection must be established, which **node** must be used, and which **TP profile** must be executed on the host.
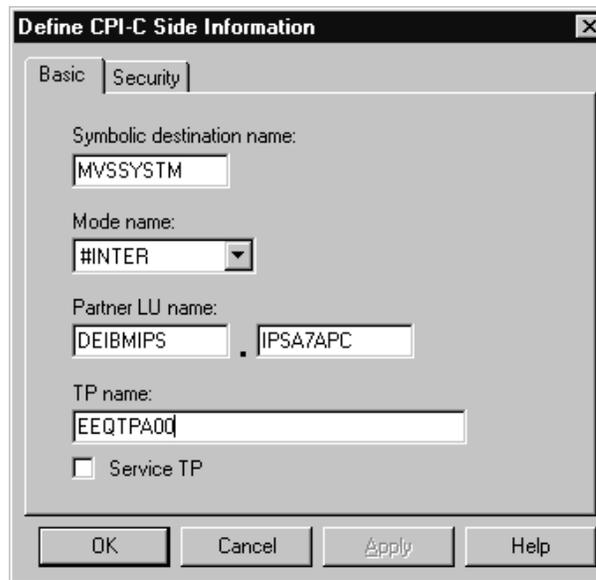
Figure 15. Define CPI-C Side Information

   b. Check the **Security** page of this dialog. You can accept the offered values and press **OK**.

8. Back in the **SNA Node Configuration** dialog, you have to save the definitions you have made. Select **Save** under the **File** menu.

9. The **Save as** dialog appears. Specify a file name. You have to open this file again when you start APPC. After saving the file, confirm that this configuration is set to be the default one.

Exit to the **SNA Node Operation** dialog. You have made all definitions but should check them now.

Checking your APPC definitions

The following procedure describes how to check your APPC definitions.

1. After you have saved your definition your are back in the **SNA Node Operation** dialog. Press the green button in the upper left corner or select **Start mode** under the **Operations** menu. The **Open** dialog appears.
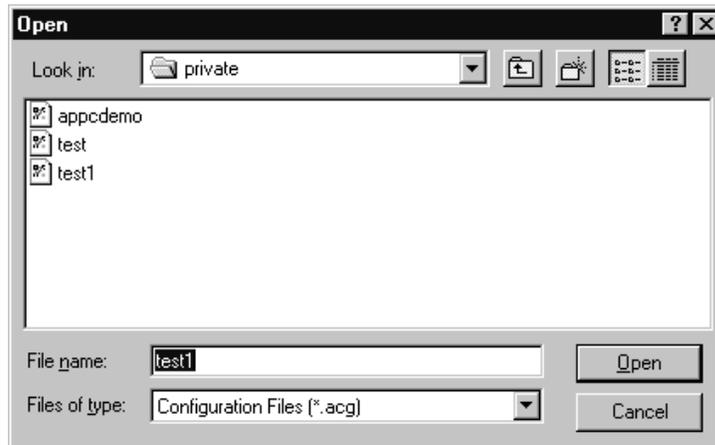
Figure 16. Open

2. Select your configuration file and open it. The **Personal Communications SNA Node Operations** dialog appears. In this dialog you can check whether your link is in active state. Specify Connections in the prompt of the title bar.
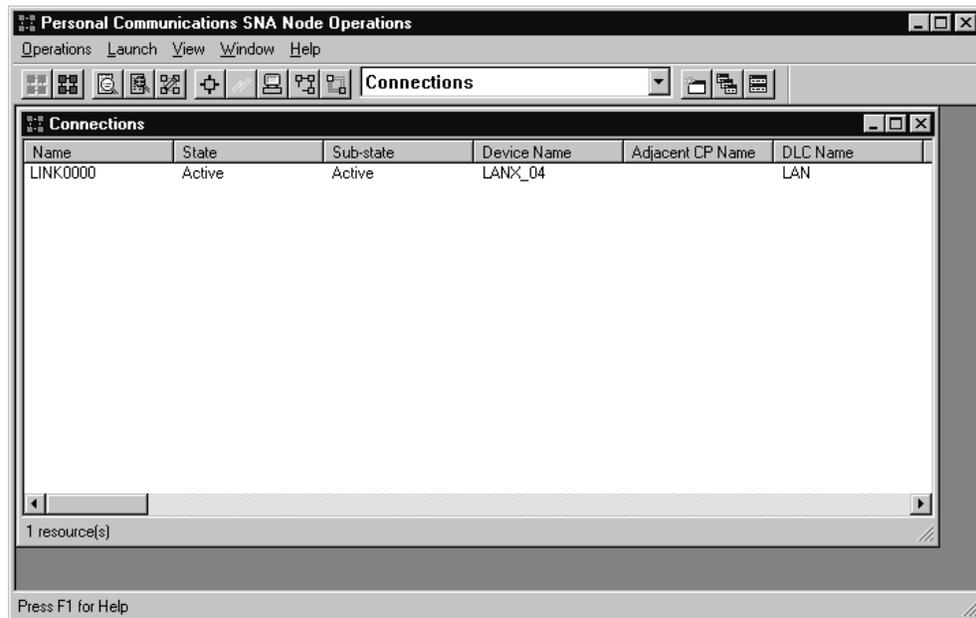
Figure 17. Personal Communications SNA Node Operations

3. To check whether sessions could be established, specify LU6.2 sessions in the prompt of the title bar and select **CNOS Initialize** under the **Operations** menu. Press **OK**.
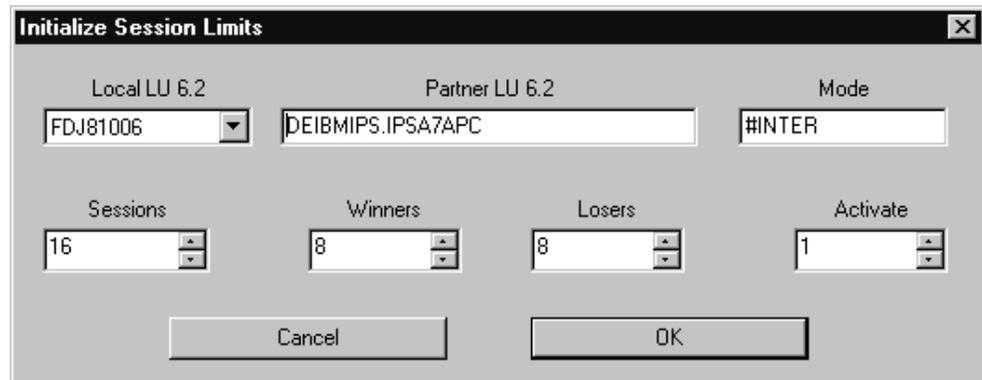
Figure 18. Initialize Sessions Limit

4. Back in the **Personal Communications SNA Node Operations** dialog select **Refresh** under **View** to show the LU 6.2 sessions.
5. If you do not get a successful connection to the host you can check for an explanation by selecting **Display SNA Sense Data** under **Administration and PD Aids**.

You can also use the APING utility to verify that all definitions in the network have been defined correctly. Select **Check Connection APING** under **Personal Communications / Utilities / APPC and CPIC utilities**. Note that you have to install some support on the host system before you can use APING successfully. See the respective product documentation for more details.

Starting APPC

Before you can invoke HCM you have to start APPC. A simple way to start APPC is to create a short-cut for this action. Invoke the CSSTART.EXE of Personal Communication. See the Windows documentation for information on how to create a short-cut.

Setting up TCP/IP definitions for z/VM

Before you can use HCM with TCP/IP, the TCP/IP networking protocol must be installed and configured on your workstation. See your TCP/IP documentation for information on how to do that.

The HCM graphical user interface uses the z/VM HCD TCP/IP dispatcher for HCM to route incoming HCM requests to HCD on z/VM. For this communication, you must provide a certain setup as described in the following sections:
- "Setting up the service virtual machine" on page 26
- "Setting up the z/VM HCD TCP/IP Dispatcher for HCM" on page 28

How to invoke HCM as the interface to z/VM HCD is described in "Starting and stopping HCM for z/VM" on page 40.

The z/VM HCD TCP/IP dispatcher is based on the Reusable Server Kernel (RSK) as described in the *z/VM Reusable Server Kernel - Programmer's Guide and Reference*. For more detailed information about the purpose of the z/VM HCD TCP/IP dispatcher, refer to the description of z/VM HCD in *z/VM: I/O Configuration*.

Setting up the service virtual machine

HCM on the workstation communicates with z/VM HCD on the host using the z/VM HCD TCP/IP dispatcher, which is also located on the host, as a mediator, using TCP/IP. The z/VM HCD TCP/IP dispatcher and z/VM HCD must run on the same service virtual machine (SVM). Exactly one dispatcher SVM is needed for each HCM client program which wants a connection to z/VM HCD.

A complete SVM for the z/VM HCD TCP/IP dispatcher for the default user ID CBDIODSP is shipped with z/VM. To set up your own customized SVM, perform the steps described in the following subsections:

- "Required minidisks"
- "CP directory entry for service virtual machine" on page 27

Required minidisks: Most of the files needed or generated by the z/VM HCD TCP/IP dispatcher and z/VM HCD may reside on minidisks as well as in shared file system (SFS). There are only two file types which must be located on a minidisk:

- the IODF file(s) used by z/VM HCD
- the trace file(s) generated by z/VM HCD

Most of the files needed or generated by the z/VM HCD TCP/IP dispatcher and z/VM HCD are rather small. Files of the following types, however, may grow very large, depending on your I/O configuration and the used trace settings:

- the IODF file(s)
- the trace file(s) generated by z/VM HCD
- the trace file(s) generated by the z/VM HCD TCP/IP dispatcher

Typically, you will maintain several IODF files reflecting different I/O configurations at the same time. For this reason, you may consider to use a separate minidisk for your IODF file management.

A trace file generated by z/VM HCD will use a maximum of 20% of free space on the minidisk. If this amount is exhausted, z/VM HCD wraps around and starts to write records into the trace from the beginning. Ensure that enough minidisk space is available if you expect large z/VM HCD trace files.

A trace file generated by the z/VM HCD TCP/IP dispatcher is not written in wrap-around mode, but will use all available space. Note that this file may reside in minidisk or SFS. Ensure that enough space is available if you expect large z/VM HCD TCP/IP dispatcher trace files.

The following tables provide an overview of required and optional minidisks.

Table 3. Required Minidisks

Minidisk	Description
4VMHCD40 400	z/VM production build/code disk
MAINT 190	Production disk containing the RSK American English message text file and the RSK run-time environment
MAINT 19E	Production disk containing z/VM system

Table 4. Required Minidisks for Installation

Minidisk	Description
4VMHCD40 2C2	z/VM HCD sample file and user local modification disk
MAINT 193	Production disk containing Reusable Server Kernel (RSK) library files

Table 5. Optional Minidisks

Minidisk	Description
MAINT 19D	Production disk containing American English help files for z/VM HCD
TCPMAINT 592	Production disk containing TCP/IP client code. These clients are useful for testing and troubleshooting TCP/IP connection problems. So this disk must be accessed if you want to use any of these programs: *ping, traceroute (tracerte), telnet, nslookup, netstat, hometest, ftp.*

CP directory entry for service virtual machine: In order to enable an HCM client program to communicate with z/VM HCD, you need one CP directory entry for each SVM introduced in "Setting up the service virtual machine" on page 26. The following example shows parts of a default CP directory entry as shipped with the product. If you want to use different settings, change according to your needs. CP directory entries can only be updated by a z/VM system administrator with the required authority.

```
1  USER CBDIODSP <password> 96M 128M BCFG
2  INCLUDE IBMDFLT
3  MACH XC
4  OPTION MAXCONN 64 QUICKDSP SVM APPLMON DIAG88
5  XCONFIG ADDR MAXNUMBER 5 TOTSIZE 5G NOSHARE
6  MDISK 0191 ... adapt to your needs
```

The values specified in this example are recommended.

The statements are explained in more detail:

1 In our example, the user on the service virtual machine has the user ID CBDIODSP. When creating a new user ID, the second parameter (`password`) defines the password to be used.

96M means that the virtual machine has 96M of primary address space available after logging in. This value is recommended if large IODFs have to be processed.

128M means that the user on the virtual machine is allowed to enlarge the primary address space size up to 128M (using the CP command `DEFINE STORAGE`).

BCFG means that the user ID `CBDIODSP` is allowed to enter CP commands of privilege classes B, C, F and G.

You can find a list of privilege classes in *z/VM CP Command and Utility Reference*. How privilege classes are used in z/VM HCD, is described in *z/VM: I/O Configuration*.

2 Includes the profile IBMDFLT.

3 The dispatcher virtual machine is restricted to run in an XC virtual machine which simulates ESA/XC architecture. This is why the 'MACH XC' entry is required.

4 MAXCONN 64 specifies the maximum number of IUCV and APPC/VM connections allowed for this virtual machine. The TCP/IP line driver of the

Reusable Server Kernel (RSK) needs this value to be set high enough. See *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for details.

QUICKDSP causes a virtual machine to be added to the dispatch list immediately when it has work to do, without waiting in the eligible list.

SVM specifies that the virtual machine is a service virtual machine. The monitor data records associated with this virtual machine include the SVMSTAT setting. The only purpose is to allow products that process monitor data to report on service virtual machines separate from end-user virtual machines. No other operations, such as transaction or wait state classification, are affected by this operand.

APPLMON: The RSK uses application monitoring. Option APPLMON is only needed if application monitoring is desired. APPLMON authorizes the virtual machine to issue DIAGNOSE code X'DC' by which an application may declare or delete a buffer for CP monitoring. After the buffer has been declared for monitoring, all data in it is collected by CP into monitor records at each sample interval. For more information about DIAGNOSE code X'DC', see *z/VM: CP Programming Services*. You can find further useful information in *z/VM Reusable Server Kernel - Programmer's Guide and Reference*, chapter *Monitor Data* and chapter *Command Description*.

DIAG88 specifies that the virtual machine is authorized to use DIAGNOSE code X'88' to validate user authorizations. For information about diagnose code X'88', see "Setting up Security" on page 32 and refer to *z/VM: CP Programming Services*.

5 The XCONFIG statement specifies control parameters for the extended configuration facilities provided in the XC virtual machine architecture. The XCONFIG ADDRSPACE operand specifies the maximum number of nonprimary address spaces (sometimes known as data spaces) and the total size in bytes of all nonprimary address spaces that the virtual machine can own simultaneously.

MAXNUMBER 5 specifies the maximum number of nonprimary address spaces that this virtual machine can create and have existing concurrently.

TOTSIZE 5G specifies the maximum total size of all address spaces that this virtual machine can create and have existing concurrently, in our example, 5 gigabytes.

NOSHARE tells CP that the virtual machine is not allowed to make its address spaces available for access by other virtual machines.

6 MDISK 0191 ... defines an A disk.

Setting up the z/VM HCD TCP/IP Dispatcher for HCM

After the previous preparations performed by the z/VM system administrator, you now need to perform the steps described in the following sections to set up the z/VM HCD TCP/IP dispatcher on your z/VM machine:

- "How to set up required files" on page 29
- "Setting up Security" on page 32

Table 6 on page 29 provides an overview of files that you need to prepare for this purpose.

Table 6. Quick File Reference

File Name	Description	
PROFILE EXEC	A REXX EXEC which is executed upon login to the z/VM HCD TCP/IP dispatcher.	See page 29.
User Map file (USERID MAP)	This file contains a mapping for incoming clients to user IDs used by the z/VM HCD TCP/IP dispatcher. It can be used to control access to the z/VM HCD TCP/IP dispatcher.	See page 29.
Storage Group file (SGP FILE)	The storage groups known to the Reusable Server Kernel (RSK) are recorded in this file. Although storage groups are not used by the z/VM HCD TCP/IP dispatcher, this file must be provided.	See page 30.
Authorization files	Files used to hold authorization data managed by the RSK. The RSK will create these files if they do not exist.	See page 30.
PROFILE RSK	REXX EXEC which is executed when the z/VM HCD TCP/IP dispatcher is started.	See page 30.
CBDCONF NAMES	z/VM HCD names file defining default file names and settings.	See page 31.
CBDSDISP EXEC	REXX EXEC used to start the z/VM HCD TCP/IP dispatcher.	See page 32.
z/VM HCD Profile (HCD PROFILE)	Profile defining the settings of your z/VM HCD session.	See page 32.

How to set up required files: This section handles the required files as introduced in Table 6, that the z/VM HCD TCP/IP dispatcher needs to startup and run.

PROFILE EXEC:

The `PROFILE EXEC` should contain the following statements:

- `'SET LANGUAGE ( ADD BKW USER'`: This statement makes available American English messages of the RSK.
- `'GLOBAL LOADLIB CBDMOD'`: This statement inserts the z/VM HCD load library into the library chain, so that the z/VM HCD TCP/IP dispatcher can load z/VM HCD.
- `RPIUCMS INIT`: By default, the CP security model is used. If you prefer to use RACF/VM as the external security manager (ESM), you must use this statement to establish the connection to the RACF/VM virtual machine.

A sample `PROFILE EXEC` file named `CBDSDPRF SAMPROF` is shipped with the product.

For more information about the statements required by RACF/VM, you may refer to *External Security Interface (RACROUTE) - Macro Reference for MVS*™ *and VM*.

USERID MAP (User Map File):

The user ID mapping file `USERID MAP A` contains information on which users are allowed to connect to the server over which transport mechanism. See *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for more information. The following is a sample user ID mapping file:

```
*
* User ID mapping file for RSK
*
MAP TCP *.*.* * CBDIODSP
```

This sample defines that any user connecting via TCP/IP with an arbitrary IP address is accepted and is internally mapped to user ID CBDIODSP. The z/VM HCD TCP/IP dispatcher does not use the mapped user ID. For that reason, the mapped user ID can be chosen arbitrarily.

A sample USERID MAP file named CBDSDUID SAMPMAP is shipped with the product. It is recommended to copy this file to the A disk of dispatcher virtual machine. Ensure that the name you choose for the user map file corresponds to the CONFIG UMAP_FILE setting in the RSK PROFILE. The sample PROFILE RSK expects the user map file under USERID MAP A.

In general, there is no need to change the USERID MAP sample. You may change it to restrict access to the dispatcher by specifying a list of IP addresses allowed to contact the dispatcher.

SGP FILE (Storage Group File):

The storage group file needs to be set up manually before the z/VM HCD TCP/IP dispatcher is started for the first time. This is required, although the z/VM HCD TCP/IP dispatcher does not use storage groups.

The PROFILE RSK sample contains file name, file mode and file type (SGP FILE A) of the storage group file using command CONFIG SGP_FILE. To create the storage group file for the first time, open XEDIT and type an asterisk (*) in the first column of the first line and save the file. See *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for more details.

A sample storage group file named CBDSDSGP FILE is shipped with the product. You may copy this file to the A disk of dispatcher virtual machine. Ensure that the name you choose for the storage group file corresponds to the CONFIG SGP_FILE setting in the RSK PROFILE. The sample PROFILE RSK expects the storage group file under SGP FILE A.

Authorization Files:

The authorization files need not exist upon first start of the z/VM HCD TCP/IP dispatcher. They will be created automatically. See *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for more details. According to this manual, the authorization files should be located on separate minidisks (or in SFS). Because the z/VM HCD TCP/IP dispatcher does not make use of the RSK authorization service, they can be located on the A disk without any harm.

The sample PROFILE RSK creates these files on the A disk.

PROFILE RSK:

The PROFILE RSK controls the behavior of the Reusable Server Kernel (RSK) and the z/VM HCD TCP/IP dispatcher. See *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for information on the profile.

A sample PROFILE RSK named CBDSDRPF SAMPPROF is shipped with the product. It is recommended to copy this profile to the A disk of the dispatcher

virtual machine. In general, there is no need to change this sample. It is preconfigured for storing the authorization files, the storage group file and the user map file on a minidisk using the default file names and types. If you want to store these files under SFS or want to control the file names and types you have to change the sample profile according to your needs.

> **Note:**
>
> If the name of your TCP/IP service virtual machine is not equal to `'TCPIP'`, then you have to change the TCP START command in PROFILE RSK. Refer to the *z/VM Reusable Server Kernel - Programmer's Guide and Reference* for information on the TCP START command.

CBDCONF NAMES:

A sample CBDCONF NAMES file is shipped with the product which should work for most users without modifications. Ensure that the default TCP port setting of 51107 contained in the CBDSDISP section fits with your environment and is reserved for the z/VM HCD TCP/IP dispatcher.

Therefore, the following port entries must be in PROFILE TCPIP:
```
PORT
51107  TCP CBDIODSP
51107  UDP CBDIODSP
```

where CBDIODSP is the user ID of your z/VM HCD TCP/IP dispatcher.

If more than one dispatcher shall be started working on the same IP address, they all need different port numbers.

If there is a need to adapt the settings in CBDCONF NAMES, copy the file to the `A` disk of the virtual machine. Apply all changes to the local copy of the CBDCONF NAMES file.

CBDCONF NAMES contains a section called `CBDSDISP` (`':NICK.CBDSDISP'`) which contains the following list of variables:
- `CBDTRANT:` Defines the file name and file type of the trace file written by the z/VM HCD TCP/IP dispatcher. Default: `':CBDTRANT.CBDSDISP TRACE'`.
- `CBDTRAMD:` Defines either a minidisk file mode or an SFS directory name to which the z/VM HCD TCP/IP dispatcher for the HCM trace file is written. Default: `':CBDTRAMD.A'`.
- `PORT:` Defines the TCP port to which the z/VM HCD TCP/IP dispatcher listens. Default: `':PORT.51107'`.
- `SECURITY:` Defines the security model which the z/VM HCD TCP/IP dispatcher uses for user ID / password validation. The value 'CP' stands for native CP security for user ID / password validation basing on DIAG88. The value 'ESM' means usage of an external security manager (ESM), for example, RACF/VM. Default: `':SECURITY.CP'`.
- `DSPTRACE:` Determines whether the trace written by the z/VM HCD TCP/IP dispatcher is to be switched off after successful startup. The value 'YES' lets the dispatcher continue tracing. The value 'NO' causes the trace to be switched off after startup. Default: `':DSPTRACE.NO'`.

- `TRACELOG:` Defines the file name, type, and mode of the z/VM HCD trace file. This trace file is always needed and will be filled with at least one record when HCM closes a connection to the z/VM HCD TCP/IP dispatcher. Default: `':TRACELOG.CBDMGHOM TRACE A'`.
- `MSGLOG:` Defines the file name, type, and mode of the z/VM HCD message log file. Default: `':MSGLOG.CBDMGHOM MSGLOG A'`.
- `PROFILE:` Defines the file name, type, and mode of the z/VM HCD profile. Default: `':PROFILE.HCD PROFILE *'`.

CBDSDISP EXEC:

A REXX utility named CBDSDISP EXEC is used to start the z/VM HCD TCP/IP dispatcher. CBDSDISP does not support any parameters. CBDSDISP recognizes the '(VERBOSE ' option which can be used to obtain information on variable settings and allocations of the z/VM HCD profile.

CBDSDISP is configurable in a way such that several variable settings are read from the file CBDCONF NAMES. A sample file CBDCONF NAMES is shipped with the product.

CBDSDISP expects exactly one CBDSDISP section in CBDCONF NAMES and exactly one definition for each of the variables listed above.

CBDSDISP allocates the following files which the z/VM HCD TCP/IP dispatcher needs to run z/VM HCD:
- `HCDLOAD:` CBDMOD LOADLIB *.
- `HCDTRACE:` Filename specified by the TRACELOG variable setting from CBDCONF NAMES *.
- `HCDMLOG:` Filename specified by the MSGLOG variable setting from CBDCONF NAMES *.
- `HCDPROF:` Filename specified by the PROFILE variable setting from CBDCONF NAMES *. Allocation only takes place if the file really exists. If it does not exist, the processing continues without allocating HCDPROF.

HCD PROFILE (z/VM HCD Profile):

A sample HCD PROFILE named CBDSDHPF SAMPPROF is shipped with the product and contains a profile skeleton. You may copy this file to the A disk of the dispatcher virtual machine and adapt it. It must at least contain a comment as the first line.

See *z/VM: I/O Configuration* for details on updating the HCD PROFILE.

> **Note:**
> All of the required files have been copied to the A disk of the shipped default z/VM HCD TCP/IP dispatcher service virtual machine for user ID CBDIODSP.

Setting up Security
You can exploit native CP methods to set up security or you can use an external security manager (ESM) for this purpose. Depending on your choice, read one of the following sections:

- "How to set up security with native CP"
- "How to set up security using an external security manager": In this section, RACF/VM is used as an example of an external security manager.

To set up security on your z/VM system, you need to have system administrator authority.

You can use the following z/VM HCD TCP/IP dispatcher commands (also further described in "z/VM HCD TCP/IP Dispatcher for HCM commands" on page 389) to either display your current security setup or enforce CP security or ESM security:

- `CBDIODSP SECURITY CP`: Enforces logon verification according to CP security (DIAG88 is used). This setting is default.
- `CBDIODSP SECURITY ESM`: Enforces logon verification according to ESM settings. ESM must be initialized.
- `CBDIODSP SECURITY LIST`: Prints the security model in effect for the dispatcher (CP or ESM).

How to set up security with native CP: If you want to set up security with the help of native CP, ensure that the following prerequisites are fulfilled:

- In file CBDCONF NAMES, specify the following entry:

  ```
  :SECURITY.CP              /* :SECURITY.ESM | :SECURITY.CP  */
  ```
- In the CP directory entry for the service virtual machine, specify `OPTION DIAG88` as described in **4** of "CP directory entry for service virtual machine" on page 27. Then the password specified like **1** in the CP directory entry is authorized to use the z/VM HCD TCP/IP dispatcher for HCM.

If you select this security alternative, you can skip the next section and continue to read "Verifying TCP/IP host communication" on page 37.

How to set up security using an external security manager: This section describes the interfaces used to access z/VM security-related services and the specific actions that must be taken if you use RACF/VM as external security manager (ESM).

> **Note:**
> For any ESM, in file CBDCONF NAMES, specify the following entry:
> ```
> :SECURITY.ESM
> ```

For your convenience, section "Background information for external security manager" on page 36 provides some information that may be useful for your understanding and mentions some documentation that provides further information that you may require.

Client Authentication:

> **Note:**
>
> In the descriptions that follow, it is assumed that you use the following routines or programs:
>
> - the CSL routines DMSESM and DMSPWCHK provided by IBM
> - RACF/VM as external security manager (ESM)
> - RPIVAL provided by IBM as the password validation program
> - RACROUTE provided by IBM as the request handler

If `:SECURITY.ESM` is specified in CBDCONF NAMES, the following will happen when an HCM client requests access to z/VM HCD:

- The z/VM HCD TCP/IP dispatcher will call DMSESM to obtain a security token that may be used to uniquely identify itself to the ESM.

  The security token is included on calls to DMSPWCHK. This routine uses this token to determine whether their services should be provided using native CP functions or those of the ESM.

- The z/VM HCD TCP/IP dispatcher will call DMSPWCHK to ensure that the user ID and password provided are valid. Expired passwords may not be used to access the system and the servers do not provide a mechanism for clients to change the password.

  If DMSESM returned a non-zero security token, DMSPWCHK will verify the client's user ID and password using the command RPIVAL.

Setting up security with RACF/VM: Use the following procedure to authorize virtual machines to issue RACROUTE requests. This authorization applies to all RACROUTE requests that specify RELEASE=1.9 or any later release.

You should limit the number of virtual machines that are authorized to use the RACROUTE interface on VM. The performance of RACF/VM may be affected if many virtual machines are issuing RACROUTE requests to the RACF/VM service machine.

This section gives an outline of the actions. For more information on this topic, refer to *External Security Interface (RACROUTE) - Macro Reference for MVS and VM*.

1. Identify the RACF/VM service machine to which RACROUTE requests will be sent.

   - Make sure the dispatcher virtual machine has access to the RACF SERVMACH file.

   See the *RACF System Programmer's Guide* for more information.

2. Check if RACF/VM profile protection is in effect for DIAGNOSE X'A0' subcode X'04':

   RACF/VM provides four subcodes of DIAGNOSE X'A0' that can be used by application programs for performing security functions. The z/VM HCD TCP/IP dispatcher uses one of these subcodes (subcode X'04') indirectly when calling the RPIVAL program via the CSL routine DMSPWCHK. You have to ensure that the z/VM HCD TCP/IP dispatcher virtual machine is authorized to call DIAGNOSE X'A0' subcode X'04'.

   The use of DIAGNOSE X'A0' subcode X'04' can be protected in two ways:

   - Privilege class checking
   - RACF/VM profile protection

RACF/VM profile protection will be in effect and privilege class checking will be ignored only if **all** of the following conditions are met:

- Control for DIAGNOSE X'A0' (DIAG0A0) is turned on: There is no member called "DIAG0A0/NOCTL" in the currently active VMXEVENT profile.
- A VMCMD class profile has been defined for the subcode being issued: There is a profile called "DIAG0A0.VALIDATE" in class VMCMD.
- The VMCMD class is active.

If any or more of the above conditions is not met, privilege class checking is enforced. One or more of the following privilege classes allow for using DIAGNOSE X'A0' subcode X'04': A, B, C, D, E, F.

Note: If privilege class checking for using DIAGNOSE X'A0' subcode X'04' is in effect on your system, you can skip step 3.

3. Authorization for using RPIVAL command: Give the z/VM HCD TCP/IP dispatcher permission to perform password validation using the RPIVAL command (which uses DIAGNOSE X'A0' subcode X'04').

   a. Define a profile called DIAG0A0.VALIDATE in class VMCMD if it does not exist yet. Ensure that profile DIAG0A0.VALIDATE has a UACC of NONE:

      ```
      RDEFINE VMCMD DIAG0A0.VALIDATE UACC(NONE)
      ```

   b. Allow the dispatcher virtual machine to use DIAGNOSE X'A0' subcode X'04' by giving the appropriate user or group READ access to the DIAG0A0.VALIDATE profile:

      ```
      PERMIT DIAG0A0.VALIDATE CLASS(VMCMD) ID(userid | groupid) ACCESS(READ)
      ```

      Note: If you do not give users an access authority of at least READ when creating the access list, the users will not be able to use the subcode.

   c. If the VMCMD class is not already active, use the SETROPTS commands to activate it:

      ```
      SETROPTS CLASSACT(VMCMD)
      ```

   d. Make sure that the protection of DIAGNOSE X'A0' (DIAG0A0) is active. It is active by default, if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

      ```
      RDEFINE VMXEVENT EVENTS1 ADDMEM(DIAG0A0/NOCTL)
      ```

      If protection of DIAG0A0 is not currently active, activate it by issuing:

      ```
      RALTER VMXEVENT EVENTS1 DELMEM(DIAG0A0/NOCTL)
      SETEVENT REFRESH EVENTS1
      ```

   For more information, refer to the *RACF Security Administrator's Guide*.

4. Make sure the z/VM HCD TCP/IP dispatcher virtual machine has IUCV authorization by performing one of these two steps:

   - Update the RACF/VM service machine's CP directory entry by adding IUCV ALLOW to provide global IUCV authorization, so that any user in the system can connect to the RACF/VM service machine.
   - To give IUCV authorization to a single user, update the RACROUTE issuer's CP directory entry (for example, the default CBDIODSP z/VM HCD TCP/IP dispatcher) by adding an IUCV statement that specifies the RACF/VM service machine with which the RACROUTE issuer will be communicating, for example: IUCV RACFVM PRIORITY MSGLIMIT 255.

   See *z/VM: CP Planning and Administration* for more information.

5. RACF-authorize a connection to the RACF/VM service machine with the following substeps:

   - Log on with a user ID having the system-SPECIAL attribute.

- Create a profile named ICHCONN in the FACILITY class: `RDEFINE FACILITY ICHCONN UACC(NONE)`.
- Give UPDATE access authority to appropriate dispatcher service machines:

  `PERMIT ICHCONN CLASS(FACILITY) ID(user-ID|group-ID) ACCESS(UPDATE)`

  Update access to profile ICHCONN allows the z/VM HCD TCP/IP dispatcher service machine to issue certain RACROUTE requests on VM.
- Activate the FACILITY class (if this class is not already active): `SETROPTS CLASSACT(FACILITY)`.

6. In your PROFILE EXEC (see 29), initialize the connection to the RACF/VM virtual machine by removing the comment characters in the following line:

   `/* RPIUCMS INIT */`

Background information for external security manager:

Security Interfaces:

All password validation control services are obtained through two CSL routines:

DMSESM
> Enables access to external security manager services. This routine provides information that is used with DMSPWCHK.

DMSPWCHK
> Provides password verification.

These CSL routines in turn access ESM services using the following interfaces:

RPIVAL
> A command that determines if a z/VM user ID and password are valid.

RACROUTE
> A macro that is used to verify privileges.

The *z/VM: CMS Callable Services Reference* contains detailed information on the operation of these CSL routines, including their interactions with the ESM. These interfaces provide two levels at which you can implement your own security scheme:

- You can replace one or more of the CSL routines with those of your own creation, or those provided by another vendor, or
- You can provide your own password validation program to perform the function of RPIVAL, and a RACROUTE request handler.

Log in:

When logging into the z/VM HCD TCP/IP dispatcher service virtual machine, the PROFILE EXEC from the A disk is executed. The statement RPIUCMS INIT, which should be included in the PROFILE EXEC, enables the RACROUTE macro request handler. The operation of RPIUCMS is defined in the *RACF External Security Interface (RACROUTE) Macro Reference for MVS and VM*.

Verifying TCP/IP host communication

> **Note:**
> The information contained in this section is valid for operating systems z/OS and z/VM.

If you cannot establish a connection to the host you should first check, whether you specified the correct host name in the HCM login dialog. If the host name is correct, use the **ping** command to check the network accessibility. In a command prompt window, enter the following command:

```
ping <hostname>
```

where <hostname> is your HCM host name.

If the ping command reports an error, make sure you can reach your TCP/IP name server. Enter the following command:

```
ping <nameserver>
```

where <nameserver> is your name server's IP address in dotted-decimal notation (for example, 9.164.182.32).
If this ping command also reports an error, make sure that you specified the correct IP address for the name server (provided by your network administrator) in your Windows TCP/IP configuration notebook. If you specified the name server IP address correctly, contact your network administrator to verify that all your TCP/IP configuration parameters are correct (router IP address, subnet mask, your workstation's IP address).

TCP/IP problem determination

> **Note:**
> The information contained in this section is valid for operating systems z/OS and z/VM.

HCM displays error messages if the TCP/IP connection fails. For information on error messages, see Appendix A, "HCM messages," on page 393. You may also check for documentation updates in *HCM Documentation* under *Product Updates* provided on the HCM home page in the Internet
(http://www.ibm.com/servers/eserver/zseries/zos/hcm).

If you get messages that are not listed in Appendix A, "HCM messages," on page 393, inform IBM and provide the complete and exact message text (especially the internal description and the error stack information). If possible, make a screen shot of the message and provide information about the circumstances that caused the message to occur.

Migrating from an earlier HCM release

You can keep your configuration files from earlier releases. They are automatically upgraded when you access them with the new HCM release. However, you cannot open upgraded configuration files with an earlier HCM release. Such an attempt will produce an appropriate message.

Note: The EEQHCM.INI file is kept from an earlier release and is used by the new HCM (HCM uses the EEQHCM.INI file to store information and defaults across HCM sessions).

The following describes what to do when updating from earlier HCM releases to the current release.

Migrating from OS/390 V2R7 HCM or OS/390 V2R9 HCM

If you migrate from OS/390 Version 2 Release 7 HCM or OS/390 Version 2 Release 9 HCM, there are no special migration considerations. Just proceed as described in "Installing HCM on your workstation" on page 5.

Migrating from a release prior to OS/390 V2R7 HCM

If you have installed a version of HCM earlier than OS/390 Version 2 Release 7 HCM, you first have to uninstall the older version. Uninstall the older HCM version as described in "Uninstalling HCM." Do not delete the HCM product files manually.

After uninstalling HCM, install z/OS Version 1 HCM as described in "Installing HCM on your workstation" on page 5.

If you intend to use APPC as a communication protocol, you must create a new TP profile or update your existing TP profile. For details see "Step 1: Define TP profile" on page 14

Uninstalling HCM

Like most Windows applications, you can uninstall HCM using the standard Windows uninstall feature. Open the Control Panel by selecting **Settings** and then **Control Panel** from the **Start** menu.

For Windows 95 or Windows 98, double-click the **Software** icon and choose the **Add/Remove software** tab sheet.

For Windows NT, Windows 2000 and Windows XP, double-click the **Add/Remove programs** icon. Choose the **Hardware Configuration Manager** entry in the list box and start the uninstall process by clicking the **Add/Remove...** button or the **Change/Remove** button (for Windows 2000).

If for any reason the uninstall feature cannot delete all files, you are informed about that and have to remove these files or folders manually. Your configuration files are not deleted and you can keep them for later use.

Note: The EEQHCM.INI file is not deleted when you uninstall HCM. HCM uses the EEQHCM.INI file to store information and defaults across HCM sessions.

Chapter 3. Getting started

This chapter explains how to use the HCM user interface: keyboard, mouse, configuration diagram, objects, menus, dialogs, and reports. You can tailor your view of the configuration diagram by showing or hiding selected objects; you can also zoom, scroll, and set viewing preferences or options. You can locate, order, create, delete or edit selected objects in the diagram, using menu commands and dialog options. You can also specify and print wall charts or reports.

Once you are familiar with the HCM user interface, you can start building hardware configurations for your system (see Chapter 4, "Building your configuration," on page 63).

Starting and stopping HCM for z/OS

Before invoking HCM, make sure that your APPC or TCP/IP connection to the HCD server on the host is running. Refer to "Starting APPC" on page 25 or "Alternative 1 for z/OS: Setting up TCP/IP definitions" on page 7. Then start HCM. This can be done either by starting the executable or by double-clicking with the left mouse button on an HCM configuration file in Windows Explorer. In the latter case, the configuration is loaded as soon as HCM is running. The Product Information window of the Hardware Configuration Manager (HCM) program is displayed.

Click on **OK** and the **Login** dialog will appear.
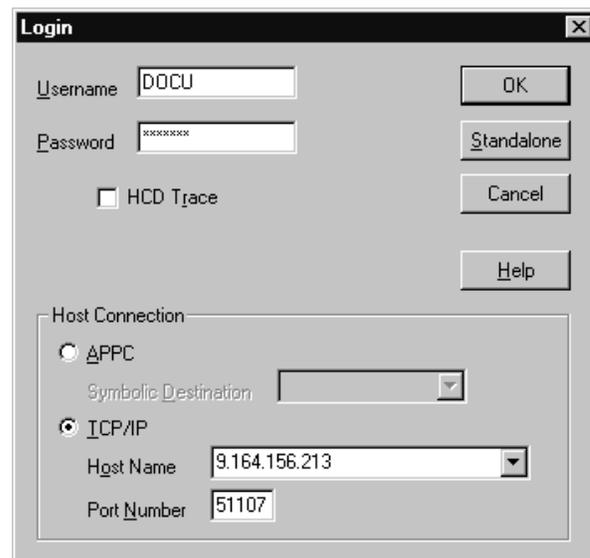
Figure 19. Login Dialog

If this is the first time you have started HCM after installing it, you must enter your user name and password and select either to use APPC or TCP/IP. If you use APPC, you have to enter the symbolic destination name for the host system. If you use TCP/IP, you have to enter a host name and a port number. The default port

number is 51107. If your TCP/IP connection uses another port number, ask your system administrator for the right number.

HCM will save the entered values in the EEQHCM.INI file, so that when you log in subsequently, you will only have to enter your password. HCM uses the EEQHCM.INI file to store information and defaults across the HCM sessions. This ini file is created *after* the first successful HCM session. It is located in the Windows system directory.

If the host machine accepts the given user name and password, the **Login** dialog disappears and you can begin your HCM session. Otherwise, a message appears, notifying you of login failure. Choosing **Cancel** will terminate HCM.

If the login fails due to an expired password, you must logon to the host via your terminal session (for example, the 3270 Emulator) to change your password. You cannot change the password in your HCM session.

If you expect an error during login, you can also click in the **HCD Trace** check box in the **Login** dialog to enable the HCD facility on the host, which traces all activity during the HCM session. Alternatively, during the session you can activate (or deactivate) the trace by toggling the **Activate HCD Trace** command in the **Options** menu. The HCD Trace is normally used only for troubleshooting HCM system problems; see "Using the HCD trace facility" on page 388 for details on how to configure the host to properly store the trace information.

Starting and stopping HCM for z/VM

Before invoking HCM, make sure that the TCP/IP connection to z/VM HCD is established as described in "Setting up TCP/IP definitions for z/VM" on page 25. Then do the following:

1. Log on to your z/VM host (service virtual machine) for which you want to use z/VM HCD. The default user ID is CBDIODSP.

2. Start the z/VM HCD TCP/IP dispatcher for HCM by issuing command CBDSDISP in CMS. Wait for text messages and verify that all services started correctly. Here is an excerpt of the messages you will see after issuing CBDSDISP:

```
IBM VM/ESA Reusable Server Kernel
(C) Copyright 1993, 1994, 1995, 1996, 1997, 1998
International Business Machines Corporation.
All Rights Reserved.

RSK>
SUBCOM   BKWSCM0000I Operation completed OK.
SUBCOM   BKWSCM0000I Operation completed OK.
...
CONSOLE  BKWCNS0000I Operation completed OK.
CONSOLE  BKWCNS0000I Operation completed OK.
...
TCP      BKWTCP0000I Operation completed OK.
CBDD200I z/VM 4.4.0 HCD TCP/IP dispatcher for HCM successfully started.
TCP      BKWTCP0000I Operation completed OK.
SUBCOM   BKWSCM0000I Operation completed OK.
CONSOLE  BKWCNS0000I Operation completed OK.
```

You may disconnect now and let the session run in the background.

3. On your workstation, start HCM as described in "Starting and stopping HCM for z/OS" on page 39.

4. Enter user ID (default: CBDIODSP) and password of your service virtual machine (SVM) into the **Login** dialog (see also Figure 19 on page 39). Use the TCP/IP host connection with the hostname or IP address of your SVM and your port number (51107 is the default).

5. Optionally you can switch on HCD trace.

6. Work with HCM.

7. Exit from HCM on your workstation.

8. Optionally, stop the z/VM HCD TCP/IP dispatcher for HCM by issuing the command SERVER STOP on your z/VM HCD TCP/IP dispatcher service virtual machine. Here is the text you will see:

```
SERVER STOP
RSK>
SERVER   BKWSRV0300I Shutdown initiated.
CBDD210I z/VM 4.4.0 HCD TCP/IP dispatcher for HCM stopped.
Ready;
```

Working with the user interface

This chapter describes how you can interact with the HCM user interface.

Keyboard

Use the function keys and special keys as shortcuts for commands. HCM offers the following keyboard functions:

F1   Opens the **Help** dialog.

F2   Shortcut for the **Fit to Window** command, which scales the diagram to completely fit in the display window.

F3   Shortcut for the most recently used named view. When selected, the last view you saved with the **Save Named View** from the **View** menu is displayed as configuration diagram.

F4   Shortcut for **Highlighted Objects** on the **View** menu. Displays only those objects selected (highlighted) in the current view and those objects logically connected to them, resulting in a *cropped view*. Settings in the *Change View Attributes* dialog are considered. That means, objects without logical connections to the selected objects may also be displayed.

Shift+F4
       Crops the view to display only objects logically connected to the selected object without considering the settings in the *Change View Attributes* dialog.

F5   Opens the **Assign Cable** dialog for the connection on the selected interface. Note that the interface at the lower end of the connection must be selected.

F8   Shortcut for the **Locate Selected Object** command. This command scrolls the most recently selected object into view – useful after creating an object.

F9   Opens the **Edit** dialog for the currently selected object.

F12   Displays the previously saved view of a configuration diagram. HCM saves a view whenever an object is included or excluded from the current view.

PAGE UP/DOWN
>Scrolls the diagram up or down a screen at a time. Can move the display to the left half of a screen at a time with CTRL+PAGE UP, or to the right half with CTRL+PAGE DOWN.

HOME
>Zooms out.

CTRL+HOME
>Zooms out all the way to **Fit to Window** (same as F2).

END Zooms in.

CTRL+END
>Zooms in all the way to **Actual Size**, which scales the diagram so that objects are viewed at their approximate size when a diagram is printed at full size.

ALT+BACKSPACE
>Shortcut for **Undo** most recent action.

DEL Shortcut for **Delete** selected object. If the selected object is a connected channel interface or a switch port connected to a CHPID, the interface will be disconnected.

Mouse

The mouse can be used to interact with the diagram in the following ways.

Click left button
>Selects the object under the mouse cursor and deselects all other selected objects. In addition, all connections to the object are highlighted. If a spanned CHPID is selected, then all instances of that spanned CHPID are highlighted.
>
>Holding CTRL while clicking selects or deselects the object under the cursor without deselecting other objects. This is used to select multiple objects.
>
>Holding SHIFT while clicking selects the object under the mouse cursor and displays only those cables connected to the selected object.
>
>**Note:** Click again without holding SHIFT to show all cables.

Click right button
>Provides a popup context menu giving access to appropriate actions on the object under the cursor.

Double-click left button
>Edits the selected object. Exception: Double clicking an interface at the lower end of a connection when the **Select Interface Option** is set to "Select Cable" will open the **Assign Cable** dialog for that connection.

Press and drag left mouse button
>Zooms in on the part of the diagram enclosed by the rectangle formed by dragging the mouse.

Press and drag right mouse button
>Zooms out so that the part of the diagram that currently fills the window will fit in the rectangle formed by dragging the mouse.

Note that certain selectable objects appear within another object in the configuration diagram, for example, interfaces and segments within a controller. To

select the controller, click on the text in the middle of the object; the entire controller and connections will highlight. To select a controller interface, click in one of the small boxes above or below which represent a channel interface or a device interface. Only the interface and its connections will highlight. Controllers such as the 3990 are divided into two or more segments; to select a controller segment, click in the segment's lower third. The segment and its connections will highlight.

Dialogs

HCM provides a series of dialogs to help you create, edit, delete, and connect objects, as well as manipulate the configuration diagram. For example, dialogs allow you to edit physical and logical information for a selected object, tailor your view of the diagram, or set parameters for a printed report.

Dialogs carry out two major functions: *navigation* and *editing.* You can navigate from a dialog to related dialogs, and edit data for a selected object. For example, you can navigate from the **Processor** dialog to the Partitions, CHPIDs and Edit Processor dialogs; in the **Edit Processor** dialog, you can edit information for the currently selected processor.

Configurations are hierarchical by nature. To reflect the hierarchy and to facilitate configuration maintenance, HCM provides an easy way to edit a lower object from a higher one by *nesting dialogs.* For example, after editing a processor, you can navigate to the appropriate dialogs to add or edit CHPIDs or partitions - without going back to the diagram.

Multiple-action dialogs include additional buttons that allow you to initiate actions without closing the dialog. If you have performed some action using these buttons that changes the data irreversibly, the **Cancel** button changes to a **Close** button. Note that the effect of the last update can be reversed with the **Undo** command.

Online help

In addition to the User's Guide, HCM comes with extensive, context-sensitive on-line help which describes each dialog in detail. Use the HCM User's Guide to find out how to do a particular task (for example, 'Connecting a patchport'), or to look up overview or reference information. Use the on-line help to find dialog-specific information while you are working in HCM (for example, what the **Edit...** button in the **CHPIDs** dialog does).

Most HCM dialogs have a **Help...** button which you can press to get information specific to the dialog currently open. You can also access on-line help at any time in HCM by using the **Help** menu:

- The **Index** command displays the Index of help topics.
- The **Keyboard** command displays the Keyboard Shortcuts topic.
- The **Using Help** command displays the Windows standard 'How to use help'.
- The **About...** command displays program specific information including the latest service level.

Each help topic in the HCM help system has a non-scrollable area at the top containing the dialog title and a See Also pop-up list box. Use the scroll bar and arrows to move up and down the help topic text. Notice that most help topics have words or phrases in the text highlighted in another color. You can browse related help topics by clicking on the *jumps* (highlighted phrases with single underline) in the text or under the "See Also" list box. You can click on a pop-up

(highlighted word or phrase with dotted underline) in the text to display the definition for an HCM term. Jumps to related help topics can appear within pop-up windows.

Use the **Back** or **History** buttons in the help menu bar to retrace your path through the help topics. Use the **Contents** button to display the Index of help topics.

Navigation maps

For an overview of HCM dialog flow, click on the **Map** button in the help menu bar (or on the "navigation map" jump in the Index). The 'Navigating around HCM' help topic contains a list of different navigation maps illustrating how you can navigate to various dialogs. Click on a map jump to display that navigation map; you can then click on any box in the dialog flow to display the help topic for that dialog. You can also browse through the navigation maps by clicking on the "Previous Page" or "Next Page" jumps, or the "-continued-" box in the map.

> **Tip:**
> In the Connect Process help topic are two navigation maps explaining the dialog flow for connecting controllers to processors and connecting devices to controllers. Click on any box in the navigation map to display the help topic for that dialog. You can access the Connect Process help topic by clicking on the "Connect Process" jump in the Index of help topics.

User data fields

In addition to characterizing an object with an ID, description, serial number, type, and so on, HCM allows you to define twenty data fields unique to your installation. For example, you may wish to record information such as the lease status, corporate asset number, or physical location (building and/or data center map grid) for a given piece of hardware. Use the **Field Names...** command from the **Options** menu to assign customized names to the user data fields.

> **Tip:**
> Consider using some of the fields for global items such as asset number and physical location, and some for local items such as data usage (for example "IMS databases") for a string.

These fields can be specifically used to appear in particular reports. Any empty user field for an object will not be printed. For example, if "Usage" is used as a field for defining the data on a string, it will only appear in the String reports. The reports will be easier readable if the names are 12 characters or less (e.g. "Asset No.", not "Asset management number").

User data can be added and modified when editing an object by clicking on the **Info...** button on the edit dialogs for physical object. This opens the **User Data** dialog which shows the user field names and the related values.

Using the **Customize Field Names** dialog, you can apply meaningful names to the fields as shown in Figure 20 on page 45. You reach this dialog from the menu bar: *Options ---> Field Names...*.
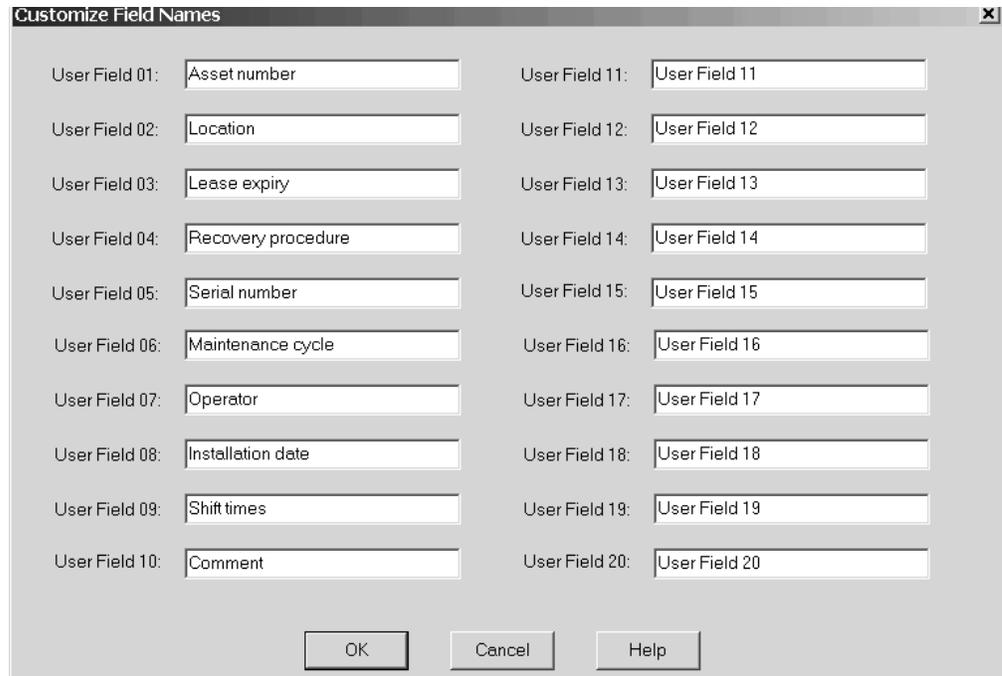
Figure 20. Customize Field Names

Messages

HCM displays HCD messages as well as HCM messages.

HCD messages

HCM displays all HCD messages that it receives in dialogs on the PWS. Whenever HCD has a single message, the **Message** dialog appears.
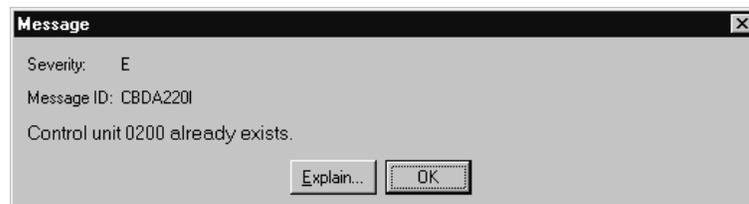
Figure 21. HCD Error Message

Click the **Explain...** button to open the **Message Details** dialog which displays more information about the selected message.

Message Details ⊠

Explanation:
The control unit specified has been defined previously.

System Action:
System waits for user action.

User Response:
Specify a unique control unit number.

OK

Figure 22. Message Details Dialog

Whenever there is more than one message, the **Message List** dialog appears.

Message List ⊠

Sev	Msg. ID	Message Text	
W	CBDG449I	Control unit 0100 of type 3990 is connected to processor PAYROLL by serial attachment, but its unit address range does not start with 00.	Explain...
W	CBDG449I	Control unit 0100 of type 3990 is connected to processor WEBSERV by serial attachment, but its unit address range does not start with 00.	Delete
W	CBDG449I	Control unit 0101 of type 3990 is connected to processor PAYROLL by serial attachment, but its unit address range does not start with 00.	Save...
W	CBDG449I	Control unit 0101 of type 3990 is connected to processor WEBSERV by serial attachment, but its unit address range does not start with 00.	OK
W	CBDG449I	Control unit 0200 of type 3990 is connected to processor PAYROLL by serial	Help

Figure 23. HCD Message List Dialog

In the **Message List** dialog click on the **Save...** push button to save the messages into a file. Before saving certain messages you can delete those messages you are not interested in.

HCD messages start with "CBD". Most HCD messages result from the user requesting a change to HCD that would create an invalid IODF. For more information about HCD error messages, see *z/OS and z/VM HCD Messages*.

HCM messages

HCM messages are displayed in *message boxes* that describe the situation and sometimes request input. Most HCM messages result from the user entering incorrect or incomplete information in a dialog. These messages are self-explanatory and are not included in the list of HCM messages in Appendix A, "HCM messages," on page 393.

Serious HCM messages are also displayed in message boxes and only have an **OK** button for you to acknowledge them. These messages start with "EEQ" and can be found under Appendix A, "HCM messages," on page 393.

Figure 24. HCM Message

Maintaining the configuration diagram

With HCM, you can build and maintain an interactive configuration diagram containing all the physical and logical information about your hardware configuration. HCM draws the objects, connections and text annotations in the configuration diagram in the clearest possible way. You don't have to worry about how the diagram will look when you create, edit or delete objects or when you filter your view; HCM will automatically redraw the diagram.

The following figure is an example of a configuration diagram.
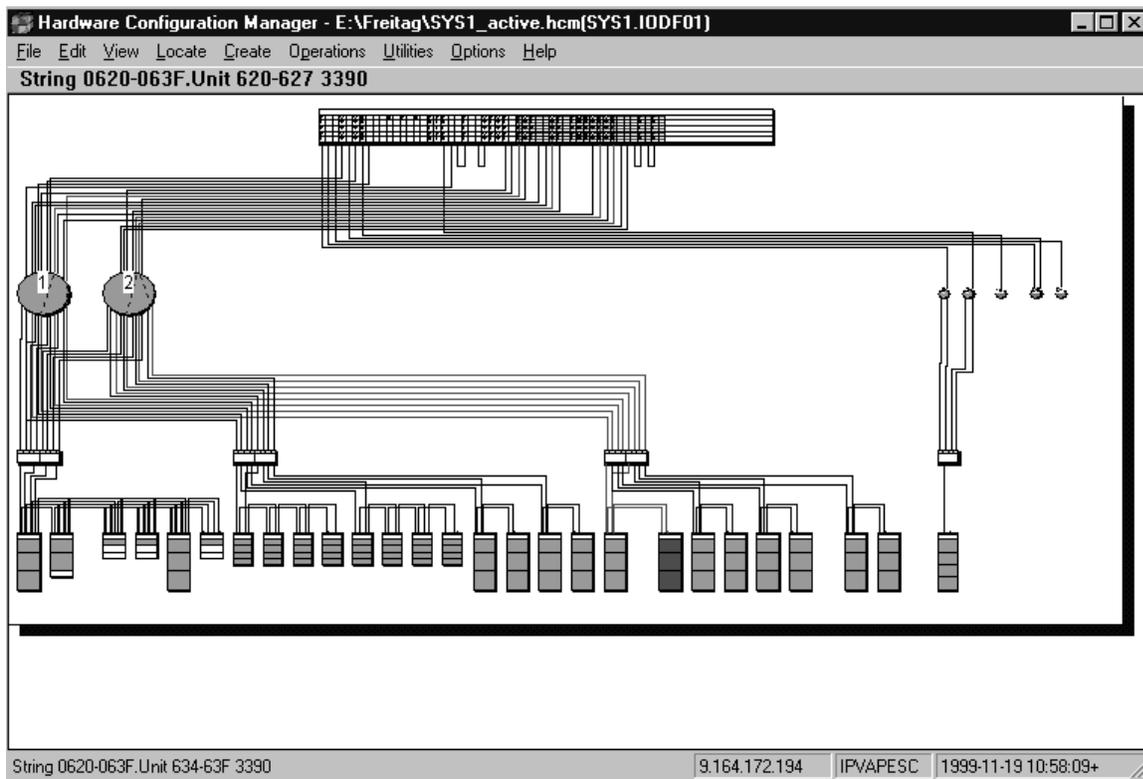
Figure 25. Example of a Configuration Diagram

Connections in the diagram

Connection lines on the diagram are drawn with different thickness, style, and graphics, with the following meanings.

Thick line	Bus and tag cable, permanent connections
Thin line	Mutable connections
Thick line with crow's feet	fiber trunk
Dashed line	Indirect connection. Some objects between the two that are connected by a dashed line are not currently included in the diagram.
Dotted line	Connection to uninstalled equipment.

When you selected to visualize a configuration, the connection lines may have a different meaning. Refer to "Visualizing ESCON or FICON switch and crossbar switch configurations" on page 51.

Certain objects such as controllers and strings can be marked as not yet installed; these objects (and cables attached to the objects) are drawn on the diagram with dotted lines running diagonally through them. To mark an object as uninstalled open the **Edit...** dialog for the object. Deselect the **Physically Installed** option by clicking in the check box, and press **OK**.

Filtering your view

You can tailor your view of the configuration diagram to focus on only the objects of interest. This results in a more readable diagram and will speed diagram drawing. Working with a subset of the configuration is easier than working with the entire diagram. Since many user actions require the diagram to be rebuilt (for example, connecting objects), you might have to zoom back into the area of interest after each of these actions. Cropping the diagram also results in better performance, since only that smaller portion must be recreated.

You can tailor a view to your needs in the following way:

Open the **Change View Attributes** dialog by choosing **Filter diagram...** from the **View** menu.
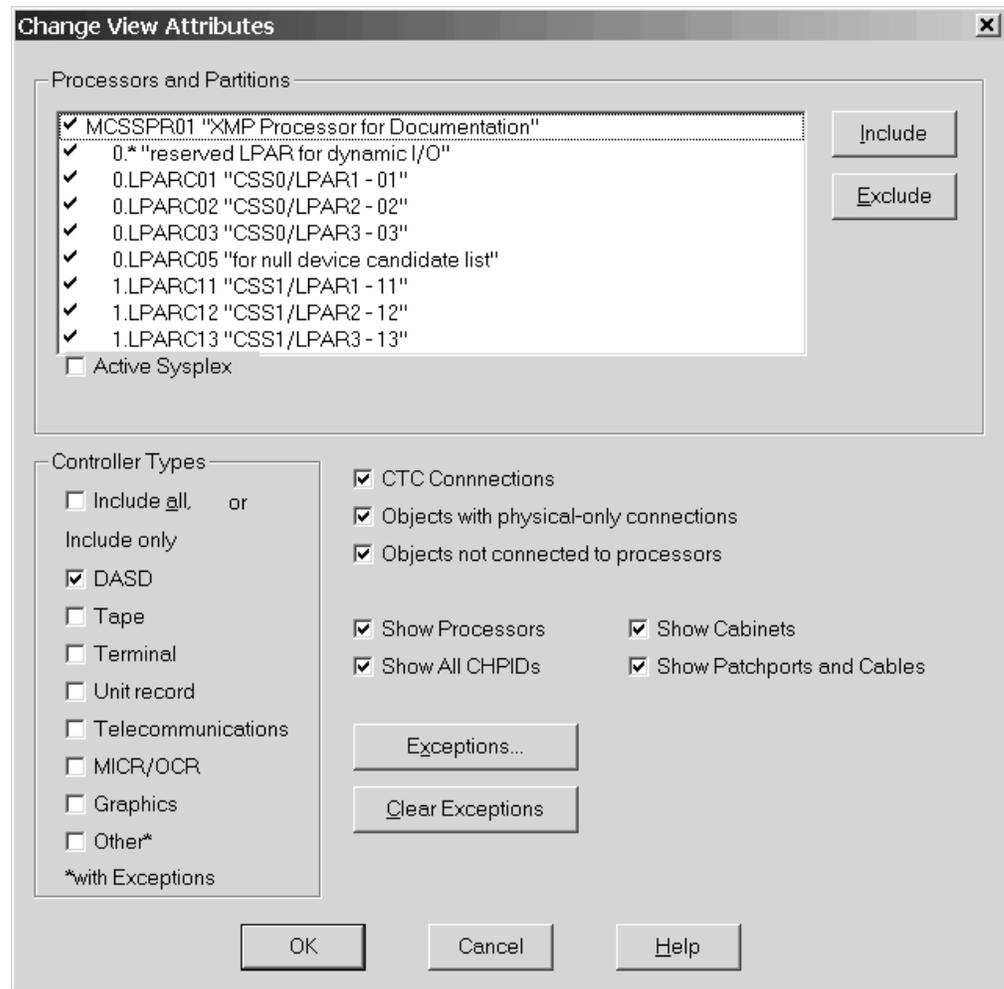
Figure 26. Change View Attributes Dialog

Use the dialog to include and exclude the types of objects you want to see. Note that you may set *exceptions* to the selected controller types in the **Change View Attributes** dialog with the **Exceptions** dialog. Exceptions allow you to include or exclude individual controllers in the class from the view.

After having selected the object types, you can now select one or more objects and press F4 (Highlighted Objects) to further reduce the diagram for displaying just those objects. Pressing F4 considers the settings made in the *Change View Attributes* dialog, while pressing SHIFT+F4 crops the diagram without considering these settings.

To return to the original diagram, press the **Clear Exceptions** push button in the **Change View Attributes** dialog. Alternatively, you can press the F12 key to restore the view that you had before pressing F4 or SHIFT+F4.

The following commands in the **View** menu help you to tailor your view:

Filter Diagram
> Allows you to specify which processors, partitions, and controllers are displayed to reduce the number of objects in the current view. The default setting for the **Include only List** is to show all DASD controllers; you can change the default and then click on **OK**. You can use the **Save Named**

View option from the **View** menu to save the current view attributes to a named view. If you save the named view, it will become the most recently used named view.

By clicking the check box **Active Sysplex** it is possible to filter the diagram to those objects, which are part of the active sysplex on which the HCD host is running.

If the Active Sysplex is selected via the checkbox, the **Show Managed CHPIDs** checkbox is activated if there are any CHPIDs that:
- are contained in one of the processors that comprise the active sysplex,
- are defined as managed, and
- have an I/O Cluster value that corresponds to the name of the active sysplex.

By selecting this checkbox, an asterisk ('*') in the diagram at the corresponding channel interface indicates that a channel path is manageable.

You can display or hide controllers by including or excluding controller display classes in the **Exceptions** dialog following Filter Diagram (see "Editing a controller" on page 116). You can also designate specific controllers to always be included (or excluded) regardless of whether their display classes are included. You can also display objects connected to processors but not to controllers, or objects not connected to processors.

The last specified filter is stored in the HCM config file across HCM sessions.

Highlighted Objects

Displays only those objects selected (highlighted) in the current view, resulting in a "cropped" view of the diagram. A *cropped diagram* is a subset of the entire configuration comprising the selected object(s) and any objects to which they connect. You can print customized cropped views as part of a report. Shortcut: press the F4 key if you want to consider the settings made in the *Change View Attributes* dialog, or press SHIFT+F4 if you do not want to consider those settings. To open another view, use the **Filter Diagram** command and include/exclude specific objects.

> **Tip:**
>
> The **Highlighted Objects** command works as a shortcut for setting *exceptions*. For example, if you have the DASD class selected in the **Filter Diagram** dialog, select a single DASD controller in the diagram, and press F4, all other DASD controllers are set to "excluded". That is, the diagram will display all controllers of class DASD *with the exception of* the DASD controllers which have been explicitly excluded. This results in a diagram containing only the selected controller.
>
> To return to the original diagram, press the **Clear Exceptions** push button in the **Change View Attributes** dialog or F12. Pressing SHIFT+F4, the settings made in the *Change View Attributes* dialog are no longer considered. This usually results in diagrams which contain even less objects than pressing F4 only. You can again return to the previous view by pressing F12 or by changing settings in the *Change View Attributes* dialog accordingly.

Saving views

You can save views you might want to display again by choosing **Save Named View...** from the **View** menu. When you want to display a saved view, you can either press (F3) to display the most recently used named view or choose **Restore Named View...** to select a view from a list box.

A useful capability of named views is to create views that show all objects connected to a particular processor or partition, possibly filtered by controller type. For example, in a configuration where processors have test and production partitions, it may be useful to create one view for production equipment and another view for test equipment.

Note: A named view saves the individual objects as they are currently drawn on the diagram. When you restore a view, it does not show objects you added after saving the view although they might belong to the same object class shown in your named view.

The following commands under the **View** menu support you in saving views:

Restore Named View
Allows you to choose a saved named view you want to display. To display the most currently saved named view you can use the shortcut F3.

Save Named View
Allows you to either save the displayed view under a new name or to update an existing saved named view.

Delete Named View
Allows you to delete one or more named views.

Visualizing ESCON or FICON switch and crossbar switch configurations

HCM can evaluate information from ESCON or FICON switch or crossbar switch configurations. After you have selected a configuration for one or more switches, HCM evaluates the configuration data and immediately shows dedicated

connections and blocked ports. In addition, when highlighting objects in a diagram, HCM uses the switch configuration data to show what particular route is used for CHPID-to-control-unit definitions. This is especially important when there are definitions in the configuration that have more than one possible physical route, for example when chained switches are used.

You can define and save several configurations and restore them whenever you want to visualize a particular configuration. Assume you have two processors that are used as a backup for each other. Therefore, you have different switch configurations defined, for normal operation and for backup operation. When you define and save both configurations with HCM, you can always view which devices are accessible when a particular configuration is active.

When you select a configuration to be visualized the following is immediately shown within ESCON and FICON switches and crossbar switches:

Solid black line	Dedicated link between the ports
Gray port	This port is blocked
Black dot	In crossbar switches, a link between a crossbar channel interface and a crossbar control unit interface

In addition, when you highlight an object, HCM evaluates the configuration data and highlights the valid path from the CHPID to the control unit. It uses the following highlighting rules:

Within ESCON and FICON switches (internal routes)	
Solid line	There is dedicated link between the ports
Dotted line	There is a dynamic allowed link between the ports. If the link is prohibited, there is no line between the ports.
For cables (Physical routes)	
Solid line	Allowed physical route.
Dotted line	The visualized configuration does not allow the physical route.

In the following example, control unit 500 is logically defined to CHPID 20 and CHPID 21 using switch 3 as dynamic switch. Without visualizing the configuration, HCM would show all possible paths through switch 4.
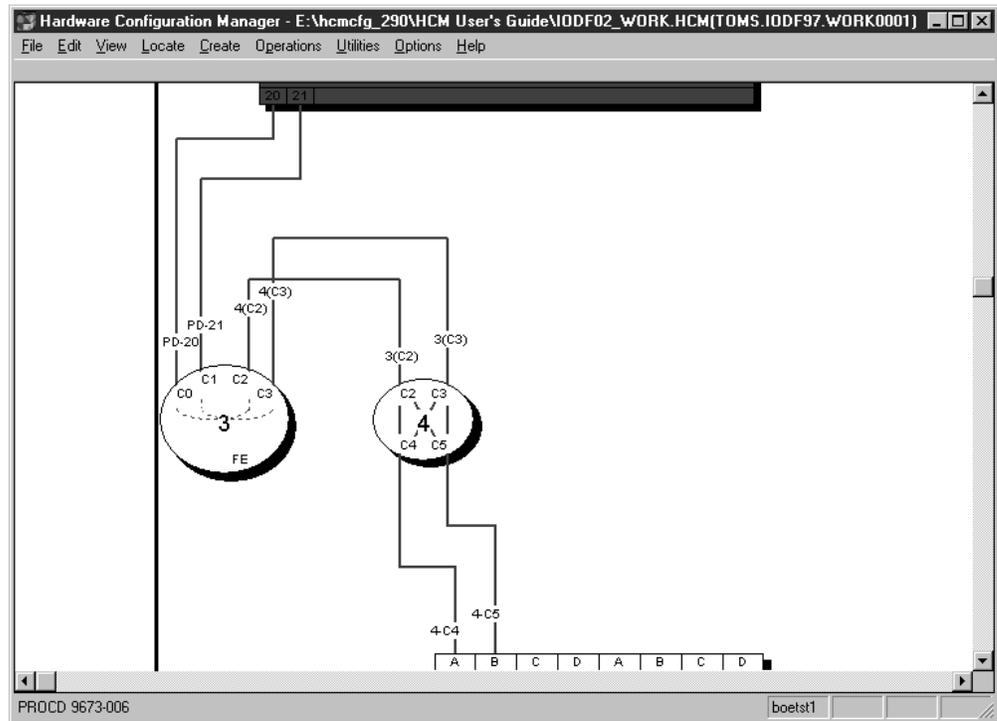
Figure 27. Highlighting an Object, without Visualized Configuration

Considering the switch configuration data of switch 4 that defines dedicated connections between C2 and C4, and C3 and C5, HCM highlights only the valid paths as shown. Note that the route in the dynamic switch is highlighted as a dotted line whereby the dedicated link is highlighted as a solid line.
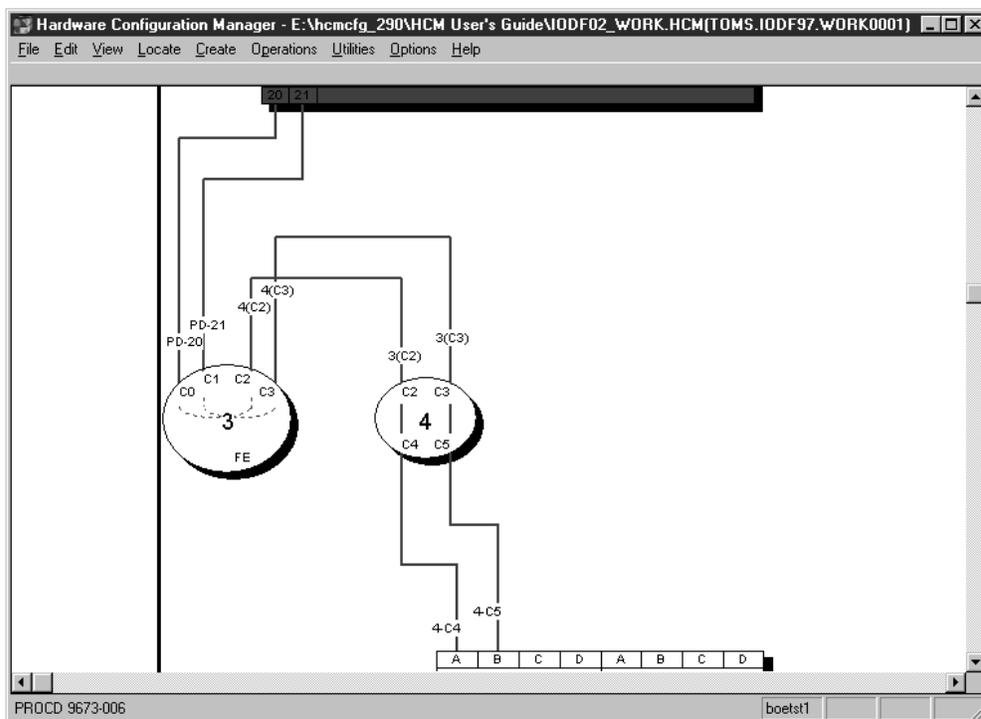
Figure 28. Highlighting an Object, with Visualized Configuration

To visualize a switch configuration, you first have to associate a particular switch configuration to each switch in your configuration using **Visualized Matrices...** from the **View** menu. The defined switch configurations are shown in your currently displayed diagram. You can save these definitions and display them later on again.

The following actions under the **View** menu support you in visualizing switch configurations:

Visualized Matrices
> Allows you to associate an active configuration for each ESCON and FICON switch and crossbar switch in your configuration.

Restore Named Matrix Set
> Allows you to choose a saved named matrix set you want to display.

Save Named Matrix Set
> Allows you to either save a current matrix set under a new name or to update an existing named matrix set.

Delete Named Matrix Set
> Allows you to delete one or more names matrix sets.

Showing page breaks

You may see unexpected gaps on the screen, particularly between CHPIDs on a processor. These gaps are needed, because HCM must be able to split the entire configuration diagram into printable pages so that each output device can be printed as a whole on one piece of paper. Choosing the **Show Page Breaks** command from the **View** menu will display the breaks between the printable pages.

Zooming the view

Other commands from the **View** menu allow you to change the size of the configuration diagram view:

Fit to Window
> Scales the diagram to completely fit in the HCM window. Shortcut: press the F2 key or CTRL+HOME.

Zoom Out
> Shrinks the diagram such that more of it fits in the window. Shortcut: press the HOME key. You can also zoom out by holding down the *right* mouse button and dragging; see "Mouse" on page 42. Note that as you zoom out, text becomes smaller and simpler, until it is no longer displayed because it is too small to read.

Zoom In
> Magnifies a portion of the diagram for a closer, more detailed look at objects. Shortcut: press the END key. You can also zoom in by holding down the *left* mouse button and dragging; see "Mouse" on page 42 above. Use the scroll bars which appear when you zoom in to navigate around the diagram. As you zoom in, text may appear as it becomes large enough to read.

Actual Size
> Scales the diagram such that objects are viewed at their approximate size when a diagram is printed at full size. *Full size*, or 100% scale, is the largest scale for printing diagrams. At this size, each controller will be about one inch tall and all text information for each object is visible. Shortcut: press the CTRL+END keys.

Scrolling the view

As you zoom in to the configuration diagram, scroll bars will appear at the bottom and right hand side of the screen. Click the arrow buttons, click in the scroll bar, or drag the scroll box along the scroll bar to navigate around the diagram. Alternatively, press the arrow keys on the keyboard to scroll the diagram up, down, left or right. To scroll the diagram half a page at a time, use PAGE UP (up), PAGE DOWN (down), CTRL+PAGE UP (left), and CTRL+PAGE DOWN (right).

Note that making or breaking a connection may move the selected object such that it is no longer visible in the current view. You may have to zoom out and then in, or scroll around to find the selected object.

Controlling text detail

You can control the text that will appear in the diagram by selecting the **Text Detail** command from the **View** menu.

Setting preferences

You can display adapters on controllers grouped by segment or adapter, using the **Preferences** dialog from the **View** menu. You can also specify the shape for converters or compress the diagram horizontally. The settings you select will be saved for future HCM sessions.

Grouping Adapters on Controllers
> For controllers that have channel adapters, use this option to choose whether channel adapters (and device adapters) are grouped on the diagram by adapter or by segment. For example, a bus architecture

controller may have two buses, 0 and 1 (represented by segments in HCM) with four adapters each, A,B,C,D. Choosing **Group by Adapter** will order the adapters: 0A,1A, 0B,1B, 0C,1C, 0D,1D. Choosing **Group by Segment** will order the adapters: 0A,0B,0C,0D, 1A,1B,1C,1D.

Long Hexagon / Pentagon
Display all converters in the diagram as either elongated hexagons or pentagons.

Compress diagram horizontally
Compresses the width of objects to result in a more concise diagram. Less paper will be used to print the diagram, but cable descriptions may overlap, making them difficult to read.

Showing screen elements

You can show or hide elements of the HCM screen with the following commands from the **View** menu:

Object Information Bar
Shows or hides the Object Information Bar at the top of the screen. This bar contains information about the objects and their connections currently under the mouse cursor. Moving the cursor across objects causes this information to constantly change.

Status Bar
Shows or hides the Status Bar at the bottom of the screen. This bar contains information about the most recently selected object as well as miscellaneous messages. If you have selected more than one object, the information shown pertains to the last one selected. In addition, you can find information about the Login target, Active Sysplex name, Focus host and Time of last Refresh.

Show Page Breaks
Shows or hides the page breaks that occur when the diagram is printed at full size. The white space between each pair of lines represents areas that are unprintable with the currently selected printer and paper size.

Specifying options

You can ask HCM to always confirm a **Delete** command; that is, when you delete an object or disconnect an interface, HCM will ask you if you are sure you want to delete or disconnect. Choose **Confirmation** from the **Options** menu, select the checkbox and press **OK** (confirmation "on" is the default).

HCM offers the possibility to choose the action that is performed when double-clicking on a Control Unit or String interface. You can choose to either edit the physical cable assignment for the interface or edit the Control Unit or String that contains the interface.

Showing system status information

HCM interfaces with I/O Operations on the host, which enables you to graphically view the objects in the active sysplex. For details, see Chapter 11, "User interface to I/O Operations," on page 221.

Receiving visual warnings

HCM will warn you of any likely errors between physical connections and logical definitions. The warnings take the form of coloring objects and connections in the configuration diagram. You can specify which warnings are enabled, as well as the color each warning displays.

To display physical-logical inconsistencies, objects are filled and cables are drawn with the appropriate color. If an object has more than one warning, HCM displays the highest priority warning. Note that selected objects and cables appear highlighted in the standard red color, and take priority over visual warning colors.

Note that when you run in a video mode with only 16 colors, the visual warning colors may be different for objects than for cables: objects are painted with brushes which support patterns, but cables are painted with pens which do not. You can display the **Colors** dialog by choosing **Colors...** in the **View** menu. Use this dialog to customize the visual warnings for a given workstation.

Note: You can use the visual warning colors, for example, after a loading or a resynchronization process. This has an impact on the performance. For information on the resynchronization process, see "Resynchronizing configuration files and IODFs" on page 68.

Printing the configuration diagram

You can print the entire configuration diagram or only those objects in the current view (see "Filtering your view" on page 48). Choose **Print Diagram** in the **File** menu to scale the diagram, calculate the number of pages that will print at the selected scaling factor, or fit the diagram to print on one page. See Chapter 12, "Reports and diagrams," on page 227 for more details.

Color printing is also supported.

Selecting objects

Click to select an object in the configuration diagram; the object and all its connections will highlight, and the **Edit** and **Locate** menus automatically list the object in temporary menu commands. You can SHIFT+click (left) to highlight and display only those cables connected to the selected object to simplify your view (click to re-display all cables).

> **Tip:**
> Since objects in the diagram can be quite small, zoom in to the area of interest before selecting the desired object. You can also identify the object using the **Locate** menu or the Object Information Bar at the top of the screen.

Some objects are contained within other objects: for example, controllers contain segments and switches contain switch ports. To select a controller, click in the middle of the area below the channel interfaces. To select a segment, click in the region just above its device interfaces. To select a switch, click on it anywhere except on a switch port. To select a switch port, click immediately inside the switch adjacent to its connection, or on its connection up to the first bend. In general, you can select objects by clicking on their text labels.

In the configuration diagram, each logical instance of a spanned CHPID (see "CHPID operation mode" on page 105) is shown in the channel subsystem it

belongs to. This means that if a CHPID is spanned over three channel subsystems, for example, then three single CHPIDs - one in each channel subsystem where it is spanned - will be shown. Selecting or locating one out of the three logical instances of the CHPID from our example, selects or locates all logical instances in the diagram at once.

Actions such as delete or locate act on the most recently selected object. Note that you can *highlight* the connections for more than one object at a time by using CTRL+click (left). However, highlighted objects are not multiple *selected* objects, so you cannot delete or locate more than one object at a time.

Locating objects

In a large configuration diagram, it may be difficult to find the object in which you are interested. Use the **Locate** menu to quickly find an object. When you select a type of object from the menu, a dialog appears listing objects and their attributes. You can select one or more objects in this list. The selected object or objects will be made visible and highlighted in your configuration diagram

Figure 29. Selection List for Locating Objects, in this Example: Locating Devices

Select the **Columns...** push button to choose other attributes to be shown in the dialog. You can change the width of each column by selecting the black line that separates the column headings and dragging it to the left or to the right. You can sort the order in a column from ascending to descending by clicking on the column header.

If you locate an instance of a spanned CHPID via the **Locate one or More CHPIDs** dialog, then besides locating and selecting this instance, also all other instances of this spanned CHPID are selected and included in the view. See also "Selecting objects" on page 57.

When running multiple partitions, adding the **OS Configs** column shows useful information. Columns and filter settings are saved across HCM session.

Use the **Filter...** button to reduce the list of objects in the dialog. When you press this button you will be presented with a window in which you can select the objects you wish to locate. This reduces the size of the selection list in the **Locate One or More Devices** window for clarity. Your selection will be kept across HCM sessions.

Note: After editing an object it might not meet your filtering criteria any longer. It, nevertheless, does not disappear from the object list. To update the list you have to press the **Refresh** button.

> **Tip:**
>
> You can also use the selection list with its filter and arrangement capabilities to analyze the content of your configuration.

> **Tip:**
>
> After using the edit, create, scrolling, or zooming commands, you may lose sight of your most recently selected object. To relocate it, simply open the **Locate** menu and select the object which is shown as the first item in the menu list. (Shortcut: press F8).
>
> The first menu item in the **Locate** menu is only valid, if you recently selected only one object.

The **Locate** object and **Edit** object windows contain a **Save As...** button. This button allows all the listed objects, including the displayed columns, to be saved to a file in a format that can be used by a spreadsheet program (such as Lotus 1-2-3). Together with the capabilities of the enhanced **Filter** mechanism in the **Locate** and **Edit** menu, this allows HCM configuration reports to be created according to criteria specified by the user.

Creating objects

You can create new objects in the configuration diagram using the **Create** menu. The **Create** menu has individual menu items for each type of object.

All other objects are created by editing the object that contains them. The main dialog for the containing object allows you to navigate to other dialogs in order to create the contained objects. For example, you can open the **Partitions and CHPIDs** dialogs from the **Processor** dialog in order to add Partitions and CHPIDs to the selected processor.

You create an object by filling in details in the appropriate **Create...** dialog and pressing **OK**. See the details for the specific **Create...** dialog in Chapter 6, "Modifying objects in your configuration," on page 97.

Editing objects

You can edit an object by clicking on it in the diagram, then selecting the customized **Edit:***object* command from the **Edit** menu (or press F9). Shortcut: double-click on the object in the diagram. Alternatively, you can choose the object's type from the **Edit** menu. A list of objects appears, that you can filter and rearrange in the same way as described in "Locating objects" on page 58.

For more details, see the section for the specific **Edit...** dialog in Chapter 6, "Modifying objects in your configuration," on page 97.

The **Interface Selection...** command under the **Options** menu allows you to select which type of dialog will appear when you select an *interface* to edit: you can either edit the physical cable assignment for the interface, or edit the object which contains the interface.

Edit dialogs often lead to related dialogs allowing you to connect objects. See the section for the specific dialogs in Chapter 6, "Modifying objects in your configuration," on page 97 for more details.

Deleting objects

To delete one or more objects, click to select them in the diagram, and choose **Delete** from the **Edit** menu. Shortcut: press the DEL key after selecting the object. HCM will first disconnect the object and then delete it. If the **Confirm on Delete** option is on, HCM will ask you if you want to delete the object before actually deleting it. See "Specifying options" on page 56 for more details.

You can use the **Undo** command immediately after a **Delete** to reverse it.

> **Note:**
>
> Most subobjects, such as partitions, segments, and interfaces (including paths) cannot be deleted from the diagram. You can delete subobjects by editing their containing objects. For example, to delete a CHPID, double-click on the appropriate processor, click on the **CHPIDs...** button, select the CHPID from the list box displayed, and press the **Delete** button.

Selecting an interface and pressing the DEL key will only disconnect it. This only applies to interfaces at the bottom of a connection: channel interfaces, string interfaces, crossbar channel interfaces, and upwards-facing switch ports.

Strings and units that are integrated into a controller are not visible in the configuration diagram. They can only be accessed through the **Edit** menu or in the **Controller** dialog. Select a string and press the **Delete** button in the Strings section of the **Controller** dialog; HCM will display a confirmation dialog before deleting the string.

Reversing an action

The single level **Undo** command will reverse or "undo" your most recent action in HCM. Select the command from the **Edit** menu. If the previous action cannot be undone, the command changes to **Can't Undo**.

Ordering objects

You can order objects on the screen and in reports by using the **Order** dialog. Click on the **Order...** button in the appropriate dialog. For processors, select the **Order Processors...** command from the **Edit** menu. The standard order dialog will appear with two list boxes, one showing the objects in the current order, the other showing the objects in the new order after they have been rearranged with the **Add>>** button. The **Add>>** button adds an object selected in the Current Order list box to the end of the New Order list box. The **<<Remove** button removes the object selected in the New Order list box and places it back in the Current Order list box. Shortcut: Double-click on any item to move it.
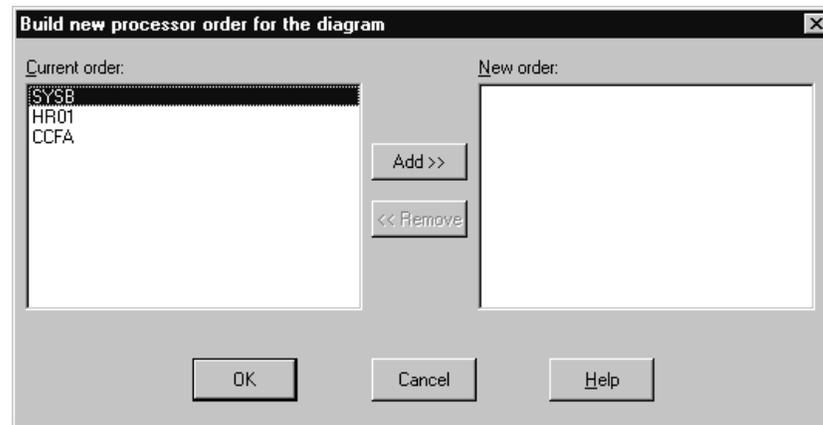
Figure 30. Build New Processor Order for the Diagram Dialog

Chapter 4. Building your configuration

This section begins by describing how HCM works with HCD: data integrity
between the IODF on the host and the config file on the PWS; and the IODF
load/resync process. This is followed by an explanation of physical and logical
objects in the configuration diagram. The last part of this section is devoted to
getting you started in HCM, taking you through the process of creating a new
configuration from an existing IODF. You can also create a new configuration from
scratch, open an existing configuration, or copy, save and close your configuration
or IODF.

HCD uses the IODF to store the logical configuration data defined for the
operating system and the channel subsystem. HCM, running on a PWS, acts as a
"graphical front end" to HCD: loading the IODF into HCM allows you to build an
associated, interactive configuration diagram in a config file. You can supplement
the IODF logical data with physical information about the physical hardware
aspects, such as cabinets. Each physical object is a graphical object in the diagram.
HCM keeps the IODF in sync with its associated config file as you make changes.
Due to the client/server relationship with HCD, all changes of the logical I/O
configuration are immediately written into the IODF at input time and fully
validated and checked for accuracy and completeness by HCD.

You can create a new configuration either by creating a new config file and IODF,
or by populating a new config file with data from an existing, prepared IODF
(most common case). You can open and/or resynchronize an existing configuration
by opening the config file with its associated IODF or against a different IODF. You
may wish to copy the config file and IODF to backup files before making changes;
this is useful when planning possible future alternatives to your system
configuration.

After you have created or opened a configuration, you can modify it. HCM allows
you to easily create, edit, connect, disconnect or delete objects in the diagram. (See
Chapter 6, "Modifying objects in your configuration," on page 97 or Chapter 7,
"Connecting objects in your configuration," on page 155). When you are finished,
close your configuration. This saves your changes in the config file and associated
IODF.

HCM and HCD

HCM is the graphical user interface to HCD. However, HCM runs on a
workstation and HCD runs on the host. The host systems require an internal
model of their connections to devices, but it is far more convenient and efficient for
you to maintain (and supplement) that model in a visual form. HCM maintains the
configuration data as a diagram in a config file on the workstation in sync with the
IODF on the host.

The *IODF* contains only logical definitions; the *config file* contains a subset of the
logical data from the IODF, plus physical information added using HCM. Together
the IODF and the config file data form a single data source or repository. This
gives you a complete view of both the logical and physical aspects of the hardware
configuration.

HCM interacts with HCD in a client/server relationship:

- HCD 'owns' all data stored in the IODF, and carries out all validation of that data. Error messages relating to the IODF come from HCD. Note that an IODF created in HCM is not different from an IODF created in HCD; no HCM data is stored in the IODF.
- HCM 'owns' the data in the config file; all validation of the physical data and error messages relating to that data come from HCM.

Each config file is associated with a single IODF; think of the config file as an extension to the IODF. When you open the config file, its associated IODF is automatically opened. You can open the config file against a different IODF, for example, if the associated IODF was damaged or renamed in HCD.

You can have multiple config files associated with a single IODF. However, this is not recommended. Every time the IODF is updated through one of its config files, all the rest must resynchronize with the IODF the next time they are opened (see "Data integrity between IODF and configuration file" on page 67).

HCM works with one IODF at a time. Use a single IODF in your data center to make the most effective use of HCM. Note that you can merge IODFs.

HCM can be used with work IODFs or production IODFs, but can only update work IODFs. A *work IODF* can be modified and updated; it serves as your 'working copy'. A *production IODF* cannot be updated; it is the machine-readable IODF used by z/OS and z/VM to define the system configuration. (You can copy a production IODF to a work IODF, or create a production IODF from a work IODF; see "Copying your configuration" on page 86).

You may wish to backup the IODF and config file before making changes. Use the **Copy to Backup** command under the **File** menu to copy both files and leave the originals open for you to work on. To only copy the IODF, use the **Copy...** command on the **IODFs** dialog. See "Copying your configuration" on page 86 for more details.

Relation of configuration files and IODFs

With HCM you can create *work configuration files* (with the file extension *.HCM) and *production configuration files* (with the file extension *.HCR). These configuration files are associated with the IODF data sets, forming a work or production pair. Figure 31 on page 65 illustrates the relationship.

Work
IODF

Production
IODF

Build production IODF

Build work IODF

Host

Associated

Associated

PWS

Build production
configuration file

Build work
configuration file

Work
Configuration
File

Production
Configuration
File
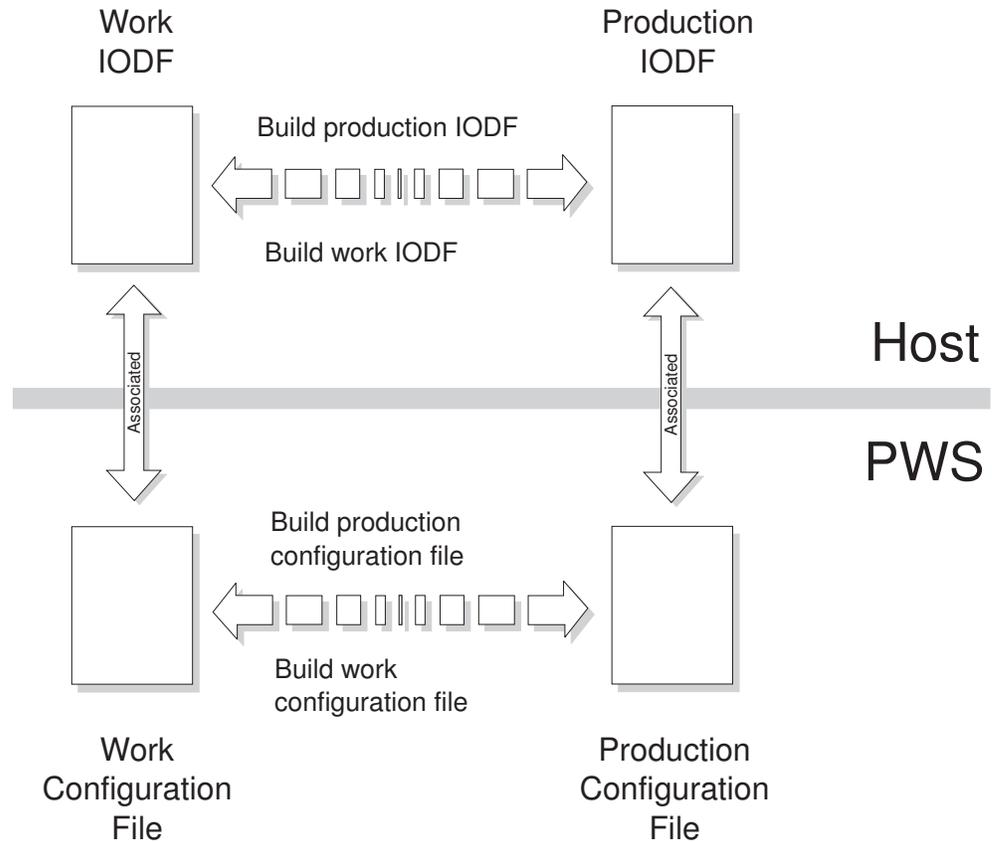
Figure 31. Configuration Files and IODFs

HCM provides menu choices for the transformation of the currently opened IODF into either a work IODF data set or a production IODF data set. You can select these choices from the **File** menu. The choices are **Build Production...** and **Make Work...**. According to the action a work configuration (*.HCM) file or a production configuration (*.HCR) file on the PWS is created.

The function **Build Production...** creates a production IODF as well as a production configuration file. These two production files are associated. That means when you open the production configuration file, the production IODF is opened as well.

When you create a production IODF, a production configuration file is created. This new file is:
• Synchronized with the host production IODF.
• Marked internally as different from a work configuration file.

The function **Make Work...** creates a pair of associated work files, the work IODF and the work configuration file.

Either a production pair or a work pair can be used as a starting point for further changes.

Characteristics of the production and work configuration File

This section describes the characteristics of the production configuration file and the work configuration file that you can create in HCM.

Production configuration file: Within an opened production configuration (*.HCR) file the physical and logical data cannot be modified. Like the production IODF, the production configuration file is a read only file for HCM. Trying to perform actions causing a configuration change results in an error message.

However, under the following circumstances HCM alters the content of a production configuration file:
- An upgrade (and potentially a resynchronization) of the file content is required. One possible reason is that the file was saved by a former release of HCM.
- A resynchronization of the file is necessary if the IODF is no longer in a state of synchronization.

A production configuration file contains the complete information that a production IODF contains and, in addition, the physical information stored in HCM.

Note: A production configuration file contains the complete information about the logical and physical configuration, but you cannot create an IODF out of a production configuration file.

This allows the production configuration file to display information on all dialogs, diagrams, and reports in the stand-alone mode. For more information refer to Chapter 5, "Standalone mode," on page 91. This information is stored in the production configuration file (as well as in the production IODF) and cannot be changed. This additional information is the reason for the larger size of the production configuration file. It is larger than the work configuration file from which it was created.

The additional information includes:
- Switch and port configurations
- Operating system related definitions
- Unit addresses and some other information from the device <=> processor and CU <=> processor relationships

If you work with HCM in on-line mode and you open a production configuration file, you can create a work IODF/*.HCM file pair from the production IODF/*.HCR file pair. To perform this task, use the **Make Work...** command in the **File** menu. See "Making a work or a production IODF" on page 83 for more information.

Work configuration file: If the open file is a work configuration (*.HCM) file, you can change the configuration. If you are in on-line mode, all physical and logical data is available in HCM. In stand-alone mode, HCM cannot report or display certain data because the data is not present in the *.HCM file. This data is only stored in the IODF data set on the host. In detail, the following data is not available in stand-alone mode:
- Switch configuration data
- Operating system related data
- Logical data of control unit - processor definitions
- Logical data of device - processor definitions

HCM configuration files

The term *config file* is used for both *work* configuration files (*.HCM) and *production* configuration files (*.HCR).

Data integrity between IODF and configuration file

The IODF is updated automatically during an HCM session. The configuration file is only updated when you save or close the configuration, or if you exit HCM. The **Save** command under the **File** menu commits any IODF changes (no longer able to Undo), and saves the configuration file. The **Close** and **Exit** commands also commit all changes. You cannot close the IODF and configuration file without committing your changes.

There are cases where the IODF gets out of sync with the configuration file; that is, the connectivity data in all configuration files associated with that IODF no longer match the contents of the IODF. See "The IODF load / resync process" for cases where the local IODF database must be refreshed from the host. HCM detects and resolves this situation by examining a last-updated specific token when the IODF is opened with a config file. HCM re-reads and resynchronizes the configuration file with the IODF if necessary.

Concurrent use by multiple users

More than one user can use HCM with the same IODF and config file, *provided none of the users is updating*. HCM initially opens the IODF and config file for read-only use, only acquiring update access for each user when required. HCM attempts to acquire exclusive update access when a user wants to update. Only one user is allowed to have access for update. If this fails, the update is refused (that is, someone else is accessing). The session continues in read-only mode.

You may grant certain users access to only the IODF or only the config file, by using the workstation and host security facilities (for example, RACF on z/OS). A user with config file access may only update some physical data (for example, configuring fiber management cabinets), but cannot alter the IODF logical data (for example, connecting controllers to processors).

To provide read access for multiple users, you can distribute a production configuration file electronically, or place the file on a central shared disk resource (for example, LAN) for common access.

Note: In case a SAMBA/SMB server is running on the host, it is also possible to provide access to the file system of USS for your workstation. You then can access the file system of USS as if it were a local drive or a LAN drive. Storing an HCM configuration file on the file system of USS means that the HCM configuration file then resides on the host. Note that when you establish the connection between HCM on your workstation and HCD on the host via TCP/IP, you always have access to USS as well as a home directory in USS. Hence, you could also save HCM configuration files in your USS directory (or any other location in the USS file system where you have access).

The IODF load / resync process

In addition to the physical data which resides only in the config file, HCM maintains a *local copy* of relevant logical connectivity data from the IODF. This local copy enables HCM to display logical information without time-consuming interactions with the host. Any changes you make are written both to the local copy and to the IODF on the host. Since the local data is saved with the config file, HCM does not have to re-read it from the host each time the config file is opened. However, there are some cases where the entire contents of the local IODF database must be refreshed from the host:

1. The first time a config file is created from an IODF (*IODF Load*). See "Loading the IODF" on page 74 for more details.
2. When the IODF on the host is modified outside of HCM by using, for example, the HCD ISPF interface. HCM will detect that the IODF on the host is out of sync with the local copy, and will re-read the local IODF data when the config file is next opened (*IODF Resync*).
3. Updating only one config file of several associated with the IODF (*IODF Load and Resync*).
4. If the connection between the workstation and host is lost during an HCM session, IODF data may be more current than config file data, since the IODF is updated automatically but the configuration file is only updated on a Save, Close or Exit (*IODF Resync*).

Generally, resynchronizing the IODF logical data has the following effect on the physical configuration:
1. New physical objects are created where appropriate.
2. New physical connections are made *as long as they do not conflict with an already existing physical connection.* Existing physical connections remain.
3. Any 'old' physical objects and connections that are no longer mentioned in the IODF remain, except for certain physical objects that depend on the existence of corresponding logical objects. For example, a physical director will be disconnected and deleted if its switch has been deleted from the IODF.

The following table gives an overview on what happens when you load an IODF or when an IODF is synchronized:

Table 7. IODF Load versus IODF Synchronize

Action	Result	More Information
Loading an IODF	Creates a new config file. If you specify an already existing config file name, all currently stored information in the existing config file is completely overwritten. It is lost.	See "Loading the IODF" on page 74
Resynchronizing IODF and config file	The local copy of all logical configuration data in the config file will be overwritten with the data of the IODF. The physical data is generally kept.	See"Resynchronizing configuration files and IODFs"

Resynchronizing configuration files and IODFs

Resynchronization between a configuration file and an IODF is a time consuming process - the definitions of both files are compared and then the configuration file is updated. HCM can resynchronize both work configuration files and production configuration files.

When you open a configuration file, HCM checks whether the configuration file on the workstation and the related IODF on the host are in sync. If they are out of sync, HCM can
- Resynchronize the content of the configuration file with the IODF on the host
- Access the content of the configuration file as read-only.

When you select to synchronize the content, the local copy of all logical configuration data in the configuration file will be overwritten with the data of the IODF. Generally, the physical data is kept.

The resynchronization operation is indicated by the following dialog:
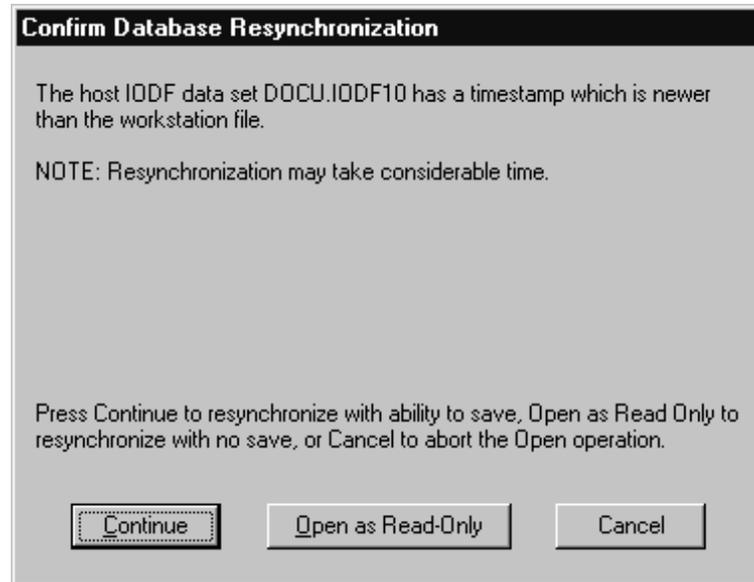
Figure 32. Confirm Database Resynchronization Dialog

If you select the **Continue** button, HCM performs the resynchronization operation. Visual warnings may pop-up to indicate inconsistencies between logical data and the physical infrastructure. Before you work with the config file or save it, resolve these conflicts. See "Receiving visual warnings" on page 57 for information on how visual warnings work and on how you can enable them.

If you select the **Open as Read-Only** button, a resynchronization takes place but the configuration file is not saved and you are not allowed to edit the configuration file - regardless whether a work or production configuration file is open.

Physical and logical objects in HCM

The IODF contains information on *logical* objects and how they are defined to each other. HCM displays *physical* objects and their connections in the configuration diagram, and contains the logical information associated with the physical objects in dialogs. For example, a device may be logically defined to a processor; however, *physically*, the device may reside in a unit in a string which connects to a controller daisy-chained to another controller, leading to a series of crossbar switches which finally connect through a CHPID to the processor. The configuration diagram shows all the connected physical objects; the device <=> processor logical definition information can be found in the appropriate dialog.

The processor only needs to know the logical routing that reaches the desired device–the logical definition. As a system administrator, you need to know how the processor and device are connected–the physical configuration–as well as the logical links.

> **Tip**
>
> Since HCM usually makes both the physical connection and logical definitions in the same step, the logical definitions will usually have corresponding physical routes. However, situations may arise where the logical definitions and physical routes do not match. HCM highlights these cases in the configuration diagram; see "Receiving visual warnings" on page 57 for details.

In HCM, most physical objects contain logical objects (for example, cabinets do not contain logical objects, but patchports, and patchports are physical objects). A logical object is contained in a physical object and in most cases, permanently associated with it. However, an operating system is a logical object, but is not contained in a physical object. Physical and logical objects are associated during IODF Load or Resync, or when the logical object is created and added to a physical object in HCM. To identify the logical objects that are associated to a physical object, the serial numbers are a major guide. For example, when assigning control units to controllers, especially in dual mode controllers.

The process of mapping logical definitions into physical objects during the Load process creates the configuration diagram in HCM. This process requires the following data:

For processors:
Any multiple processors in the IODF which belong to the same physical processor (for example, when the same processor is defined once in BASIC mode and once in LPAR mode) should have the same serial number or SNA address.

For control units:
Control units which belong to the same controller should have the same serial number (for example, the two control units in a 3990 DASD Subsystem). HCM relies on this to resolve ambiguities where other information is insufficient to make this determination.

For devices:
Devices which are physically in the same string should have the same serial number; when you have more than one device with the same device number, HCM uses the serial number to determine whether the devices are physically identical. It is usually sufficient to assign serial numbers to just the first device in each string, as devices with the same serial number will be placed in the same string. Note that if two devices have different device numbers but the same serial number, they are considered to be in the same physical string.

See "Preparing the IODF" on page 72 for details.

Note: HCM does not check if the serial numbers are accurate, but checks if the serial numbers are identical or not.

Graphical representation of physical objects

HCM represents controllers and string objects in the configuration diagram based on their *physical descriptions*. The Physical Description Files, or *PDFs*, are templates that you can tailor. They are similar to HCD UIMs, except that the PDFs contain physical rather than logical information.

HCM comes with generic PDFs for controllers and strings, as well as PDFs for selected IBM equipment. For example, the PDF for a 3990 controller specifies that it is divided into two sections, each of which contain control units. Sample PDFs are also included which may serve as examples of how you might create PDFs for non-IBM equipment.

You can modify the PDFs for objects as part of preparing the IODF for use in HCM (see "Preparing the IODF" on page 72).

Unlike HCD UIMs, you do not need a PDF for each controller and string type in your configuration. PDFs are only required in special circumstances, where the controller or string would not appear correctly in the diagram using the default PDF. For example, PDFs are required in the following situations:

1. For most DASD controllers, which typically contain two segments. (Note: segments may also be known as storage clusters, storage directors or busses, depending upon manufacturer and model.)
2. For controllers, which have a hierarchical internal architecture, such as the interfaces within channel adapters contained in busses (generically referred to as *segments* in HCM).
3. For tape controllers, such as the 3803 and 3480, which connect to peers via communicator cables.
4. For controllers and strings which number their interfaces with a scheme that differs from A, B, C...

Note: The filename extension for physical description files was changed from ".pdf" to ".txt" for HCM 2.9 Service Level 3 and higher. Nevertheless, HCM 2.9 Service Level 3 and higher can still read physical description files having the extension ".pdf".

How PDFs are assigned to objects

A controller is assigned to a particular PDF in one of the following ways:

1. When an HCM file is first created by loading an IODF (or when opening an existing file which causes an IODF resync), HCM uses the "preferred PDF" for the controllers it automatically creates. The preferred PDF for a given control unit type is determined by searching the three controller PDF directories for a PDF with a matching control unit type. The CPDFA directory is searched first, then CPDFB and CPDFC. The preferred PDF comes from the *first* directory where a match is found. If more than one matching PDF is found in that directory, the one that comes first alphabetically will be used as the preferred PDF. If no PDFs are found in any directory, HCM uses the default PDF, which is a single-segment controller with one channel interface labeled "A".
2. When a new controller is created within an existing HCM configuration, HCM will find all of the PDFs in all three directories that match the control unit type chosen by the user. If exactly one is found, then that one will be used. If two or more are found, the user will be prompted to choose the correct one. If none are found, the default PDF is used.
3. An existing controller can be edited and given a different PDF. In this case, any PDF may be chosen, even ones that do not match the control unit type.

When an IODF load is first performed, HCM (as shipped) will assign IBM PDFs to all equipment requiring PDFs, such as 3990 controllers. However, if most of your equipment is ″brand X″, or if there are multiple IBM PDFs for a given control unit type (e.g. 3990), you may wish to move the PDF best representing your equipment of that type into the CPDFA directory before loading your configuration for the first time. See Chapter 17, "Physical description files (PDFs)," on page 373 for more details.

PDFs are assigned to strings in a similar manner. PDF selection is possible during the Load IODF process and via the Import function when importing into a new config file. See Chapter 15, "Importing and exporting data," on page 317 for more information.

Creating a new configuration from an IODF

This section leads you through the process of creating a new (and possibly your first) configuration from an existing IODF. If you want to create a new configuration from scratch, see "Creating a new configuration from scratch" on page 79.

Essentially, you create a new configuration by loading data from the desired IODF on the host into a config file which is created on the PWS. However, the logical data in the IODF is not sufficient by itself to create an accurate physical representation of your hardware configuration. Although HCM can correctly infer certain physical relationships from the logical data, in general, the logical information is either lacking or ambiguous.

HCM makes extensive use of *serial numbers* added to the IODF to resolve these ambiguities and create an initial physical configuration diagram that is as correct as possible. If you have not consistently and accurately maintained serial numbers in the IODF on the host, you will need to rectify this by preparing the IODF before loading.

HCM can recognize CTC connections, if all control units (CUs) composing a CTC connection of a CNC/CTC channel pair through a switch are connected to switch ports.

Point-to-point CTC connections are recognized, if the involved CUs of a CNC/CTC channel pair have the **same** serial numbers.

Preparing the IODF

To get an accurate configuration diagram for HCM, you have to prepare your IODF before using it with HCM. If you start with a production IODF, you first have to create a work IODF out of your production IODF (see "Create work and production IODFs without the associated configuration files" on page 86). In the following description it is assumed that you use a work IODF for the preparation steps and for the loading into HCM.

If you have switches in your configuration, HCM can only represent connections of control units to channels of processors in an adequate manner, if the control-unit-to-port connections and the port-to-CHPID connections are defined in the IODF. Therefore, first of all, check the IODF and define all control-unit-to-port connections and port-to-CHPID connections if not already defined. If you do not define these connections, your resulting configuration diagram will be incomplete

and you will get logical-physical inconsistencies (see "Receiving visual warnings" on page 57). It is recommended that you use the HCD ISPF dialog to define these connections in the IODF.

As already mentioned above, HCM relies on serial numbers to resolve ambiguities. Therefore, it is highly recommended that you provide serial numbers at least to those objects in the IODF for which those ambiguities can occur (see "Physical and logical objects in HCM" on page 69) and for point-to-point CTC connections. Note that you do not need to provide serial numbers for all objects.

You can add serial numbers to prepare the IODF in two ways:

1. Load the IODF as is into a config file, update the IODF using HCM, discard the config file, and then reload the IODF.
2. Use HCD to update the IODF before loading an HCM config file (see the applicable *z/OS HCD User's Guide* or the *z/VM: I/O Configuration* for details).

The first method is probably the most effective way to prepare the IODF, and is the one we will describe here. The graphical feedback provided by HCM makes it easy to identify the objects that need work. The second method is appropriate only if your IODF is very large and requires only a few serial number changes to be made.

Load the IODF as described in "Loading the IODF" on page 74. Any physical information that requires updating will be immediately and visually obvious in the configuration diagram. This makes it easy to find and correct any problems. For example, in the following diagram, control units 1500 and 1520 are assigned to separate controllers because they have no or different serial numbers. After assigning them both the same serial number and reloading the IODF, both control units are assigned to the same controller.
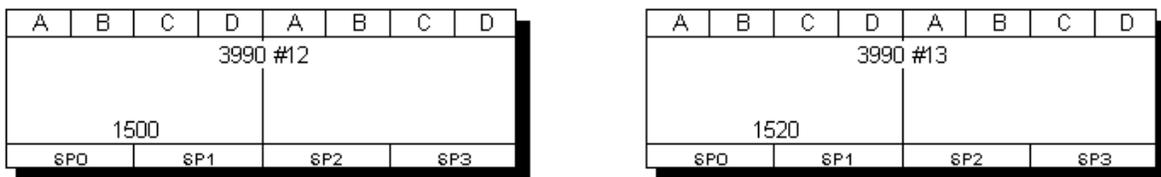
Figure 33. After loading the IODF in which control units have no or different serial numbers.
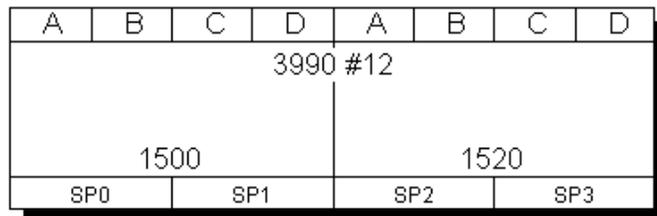
Figure 34. After loading the IODF in which control units have the same serial numbers.

Do not be too concerned if the physical configuration diagram is very inaccurate at this point. Many of these problems will be corrected when you update the IODF using HCM in the next step.

Once this first load is complete, use HCM to update the IODF by adding serial numbers to logical objects that are part of the same physical object. Double-click on an object in the diagram to display its edit dialog, and edit the serial number for the object.

See "Physical and logical objects in HCM" on page 69 for an explanation of how HCM relies on serial numbers. Pay particular attention to I/O devices which have more than one device number (e.g. a device is known as 240 from System1 and 540 from System2).

> **Tip:**
>
> It is important to correct the initial physical configuration by updating the serial numbers. Once the logical objects have been placed into physical objects, they cannot easily be moved later.

The other thing you should do before reloading the IODF is review your need for physical description files. HCM comes with a number of PDF text files which you can use and/or modify, or you can create your own. You can then select which PDFs to apply to which controllers and strings. For more details, refer to the following topics:

- "Choosing a physical description" on page 115 for controllers
- "Choose physical description" on page 126 for strings
- Chapter 17, "Physical description files (PDFs)," on page 373

See Chapter 17, "Physical description files (PDFs)," on page 373 for more details. If you have "brand X" equipment, ensure that you have read and understood the "Graphical representation of physical objects" on page 71. Controllers which have been assigned incorrectly to a PDF can always be changed later, but if most of your equipment is not well served by the installation-supplied defaults, you will find it easier to correct this *before* reloading the IODF.

After making your changes, save the config file. This implicitly saves the IODF. Then discard the config file by closing and deleting it. Reload the updated IODF into a new config file. Use the configuration diagram to check if your changes are correct. If the changes are not yet correct, change the IODF by providing the serial numbers to your logical objects again, and reload the updated IODF.

Then, you can modify the configuration in HCM, adding any desired physical information (see Chapter 6, "Modifying objects in your configuration," on page 97).

Loading the IODF

This step will load the IODF data set into a config file. If the config file already exists you will be requested to confirm that it is OK to completely overlay this existing config file.

Display the **IODFs** dialog by choosing **IODFs...** in the **File** menu.

> **z/OS:**
>
> For z/OS, the **High-level Qualifier** dialog is displayed (Figure 35). In this dialog, specify a high-level qualifier and click **OK**.

> **z/VM:**
>
> For z/VM, the **High-level Qualifier** dialog is omitted. Instead, all IODFs are displayed that are available on your accessed minidisks.
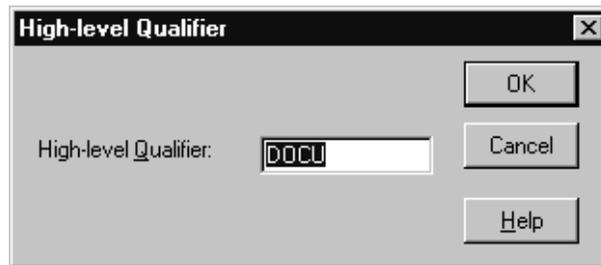
Figure 35. High-level Qualifier Dialog

The IODFs displayed in the list box in the **IODFs** dialog depend on the qualifier specified (see also "Specifying a high-level qualifier for z/OS" on page 76). The currently used high-level qualifier is always displayed below the **Qualifier...** button on the **IODFs** dialog. In the example in Figure 36 the high-level qualifier specified is "DOCU".
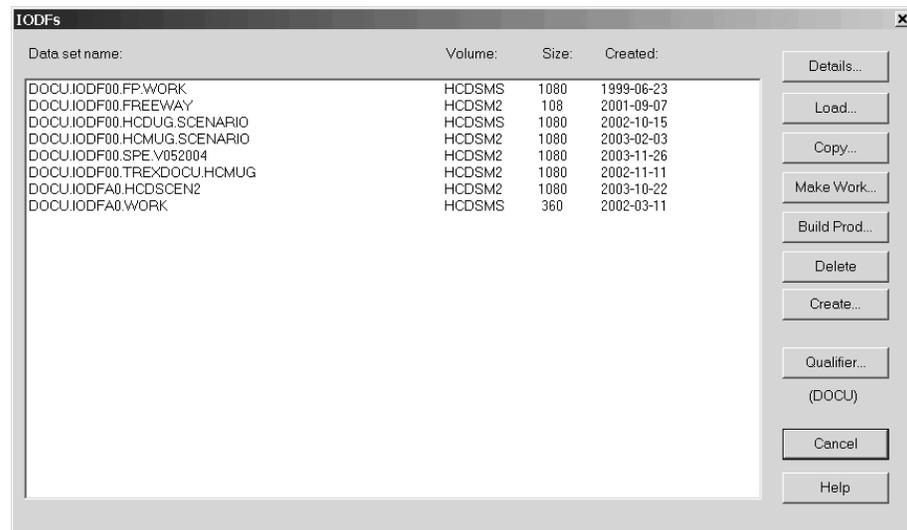
Figure 36. IODFs Dialog

Select the required IODF from the list of existing IODFs on the host, and then click on the **Load...** button to request the loading of the selected IODF configuration into the required config file.

Once the **Load...** function has started, the **Choose Local File** dialog is displayed (Figure 37). You are then taken through steps to select or create a config file and have it associated to the work IODF.

From the **Choose Local File** dialog check the HCM configuration directories selection, and change if required.
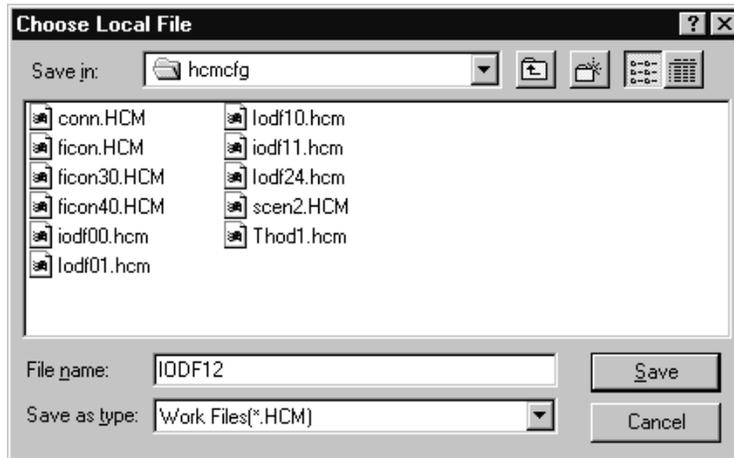
Figure 37. Choose Local File Dialog

To change the default config file directories selection, click on either the button
- In the Drives section and select the required drive or
- In the root directory and then a subdirectory entry from the selected drive.

    The drive and directory name will be saved across HCM sessions.

Tip

If you specify an already existing config file name, you are requested to confirm (Yes/No) to replace the file. If you confirm with Yes, all currently stored information in the existing config file is completely overwritten.

Since the loading process involves reading all relevant connectivity data from the IODF in order to build the configuration diagram, it can take some time, especially for large IODFs.

Specifying a high-level qualifier for z/OS

---

z/OS:

For z/OS you can click on the **Qualifier...** button in the **IODFs** dialog to open the **High-Level Qualifier** dialog. You can then specify the qualifier which determines the list of IODFs displayed in the **IODFs** dialog. The qualifier is the string of up to eight characters that precedes the first period in the data set name. Changing the qualifier will display the list of IODFs beginning with that qualifier.

```
IODFs                                                                    [X]

Data set name:                          Volume:   Size:   Updated:
                                                                          [ Details... ]
 IODF00    PRODIODF H                    HCD104    2842   2002-05-15  ▲
 IODF00    PRODIODF I                    HCD105     151   2003-07-03
 IODF00    WORKIODF A                    CBD191    8002   2004-02-11     [ Load... ]
 IODF00    WORKIODF H                    HCD104    3072   2002-05-15
 IODF00    WORKIODF I                    HCD105    4096   2003-07-03     [ Copy... ]
 IODF01    PRODIODF E                    HCD101    2222   2004-04-08
 IODF01    PRODIODF F                    HCD102    2252   2003-04-01   [ Make Work... ]
 IODF01    WORKIODF A                    CBD191    4096   2004-05-18
 IODF01    WORKIODF E                    HCD101    4096   2004-04-05   [ Build Prod... ]
 IODF01    WORKIODF F                    HCD102    4096   2004-05-17
 IODF01    WORKIODF H                    HCD104    1024   2004-05-18     [ Delete ]
 IODF01    WORKIODF I                    HCD105    4096   2003-06-12
 IODF01    WORKIODF M                    HCD107    3333   2002-10-28     [ Create... ]
 IODF02    PRODIODF F                    HCD102    2205   2004-03-17
 IODF02    PRODIODF H                    HCD104    1004   2002-02-04
 IODF02    WORKIODF A                    CBD191    4096   2004-05-24   [ Qualifier... ]
 IODF02    WORKIODF F                    HCD102    2205   2004-05-10
 IODF02    WORKIODF H                    HCD104    4096   2004-04-15        [ {} ]
 IODF02    WORKIODF J                    HCD106      80   2003-01-28
 IODF03    PRODIODF E                    HCD101    2207   2004-05-10
 IODF03    WORKIODF A                    CBD191     672   2004-04-19     [ Cancel ]
 IODF03    WORKIODF F                    HCD102    4096   2004-04-13
 IODF03    WORKIODF H                    HCD104    1024   2001-11-20
 IODF03    WORKIODF J                    HCD106     672   2003-03-18  ▼    [ Help ]
```

Figure 38. IODFs Dialog for z/VM

Choosing alternate processor definitions

If there is more than one processor defined in the IODF, HCM will ask you during
the loading process whether the processors are physically separate or *alternate
processor definitions* for the same processor.

The physical processor, known in HCM as the *machine*, can have multiple logical
processors known as *shadows*. For example, a processor may be defined once in
BASIC mode and once in LPAR mode. *The primary processor* in the machine is that
which is in the current logical view; at IODF load time, machine connections are
built for primary processors only. *Alternate* or *shadow processors* are those not in the
current view; no physical connections are built for alternate processors at IODF
load time. Every processor must be assigned to a new or existing machine. The
name of the machine is the name of the primary processor.

During the IODF load (or resync) process, you can confirm or change which
processors are assigned to which machines. The **Machines** dialog will
automatically appear when there are new processors (added to the IODF on the
host) which must be assigned to new or existing machines, or any time that some
machine has more than one processor.
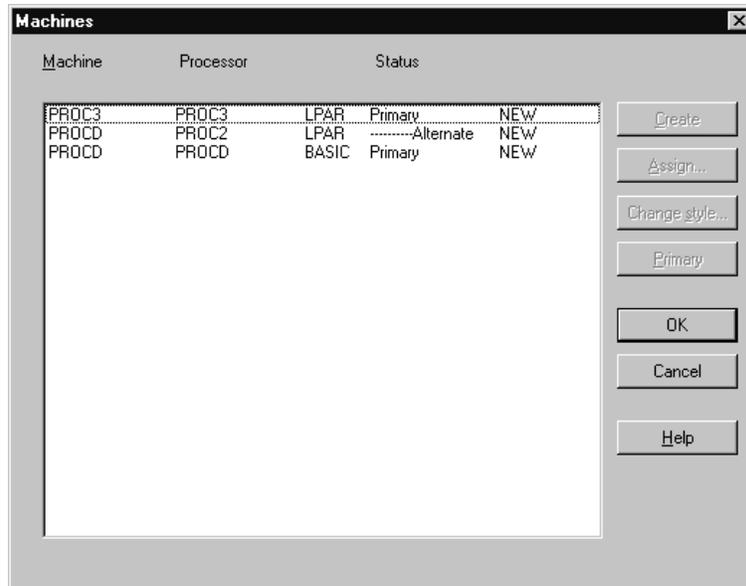
Figure 39. Machines Dialog

HCM will attempt to assign processors and primary / alternate status intelligently. The **Machines** dialog will ask you to confirm or change HCM's assignments; typically, you will press OK to confirm. For details on how to change HCM assignments, and how to manage shadow processors in general, see Appendix B, "Alternate processor definitions," on page 407.

After loading the IODF, the PWS config file and the IODF are saved automatically.

Inspecting the initial configuration

The end result of the IODF load process is the creation of an initial configuration in a config file, using the logical data and serial numbers from the prepared and loaded IODF. At this point, you can take the time to examine the configuration for any inaccuracies caused by incorrect serial numbers:

1. Ensure that you have the correct number of physical processors in the diagram. If you do not, the serial numbers are incorrect. You must re-load the IODF with the correct processor serial numbers.
2. Ensure that all control units have been placed in the correct controllers. If there are control units in the wrong controllers, the control unit serial numbers are incorrect. The control unit cannot be moved into another controller. You can either correct the serial numbers and re-load the IODF, or use HCM to delete the controller and control unit(s) and re-create and re-connect the control unit properly.

   Note: Do not worry about control units being in the correct segments, or controller interfaces being attached to the correct cables at this point.

Adding physical data to the configuration

At this point, you have a configuration that is as accurate as possible from the data available in the loaded IODF. All necessary serial numbers are available. Proceed now with updating the physical data in HCM.

Note: After you have made any physical changes, do not load the IODF again, because a load replaces the existing config file and all physical data will be

lost.

The updating of physical data is described on the following pages:

- Changing PDF assignments for controllers and strings (see "Graphical representation of physical objects" on page 71 and Chapter 17, "Physical description files (PDFs)," on page 373).
- Moving control units between controller segments (see "Arranging controller components" on page 117).
- Ordering the interfaces on controllers and strings (see "Arranging controller components" on page 117).
- Ordering daisy-chained controllers and strings in the proper sequence on their cables (see "Connecting objects" on page 156 and "Positioning objects along daisy-chained connections" on page 156).
- Creating and connecting crossbar switches and fiber management cabinets (see "Crossbar switches" on page 143 and "Cabinets" on page 146 and "Editing fiber links" on page 192 and "Connecting crossbar switches" on page 198).
- Importing physical objects, changing physical connectivity, updating descriptive fields in both the PWS file and the IODF, resynchronizing the PWS file and the IODF, (see Chapter 15, "Importing and exporting data," on page 317).
- Managing cables (see Chapter 8, "Physical connections," on page 187).

After you have made physical changes, use the **Copy to Backup** command in the **File** menu to make a backup copy of your config file.

Creating a new configuration from scratch

When building a configuration from scratch, you must create an empty IODF file as well as a new config file. From the **File** menu, choose **IODFs...** to display the **IODFs** dialog. Click on the **Create...** button to open the **Create IODF** dialog to create the new IODF on the host.

> **z/OS:**
>
> For z/OS, HCM follows TSO/E data set naming rules. In addition, the IODF name must have the following format
>
> ```
> hhhhhhhh.IODFcc.yyyyyyyy.yyyyyyyy
> ```
>
> where:
>
> **hhhhhhhh**
> is the TSO/E high level qualifier. If the name is enclosed in quotation marks then it must begin with a high-level qualifier. Otherwise, it must not and your user ID is used as the high-level qualifier.
>
> **cc**   is a set of any two hexadecimal characters (0..9 or A..F)
>
> **yyyyyyyy**
> are optional qualifiers.
>
> You must allocate enough space for the IODF. The minimum work size is 16; the default is 1024, which will allow you to define at least 4000 devices, in general. This may **not** be changed later; if the configuration expands beyond this size, the IODF must be copied to a larger data set, using the **Copy...** button on the **IODFs** dialog (see "Copying an existing IODF" on page 83).
>
> After you have entered the information, press **OK** to create the new IODF. Now select the IODF and load it as described in "Loading the IODF" on page 74.

> **z/VM**
>
> For z/VM, the name of an IODF is given by the file identifier of the CMS file. z/VM requires a certain naming convention in order to identify IODFs and accompanying files.
>
> The CMS *file name* of an IODF must be IODFxx, where xx are two hexadecimal digits.
>
> The CMS *file type* is:
>
> **WORKIODF**   for a work IODF
>
> **PRODIODF**   for a production IODF
>
> **BACKIODF**   for a backup IODF
>
> The file ID of the corresponding activity log consists of the CMS *file name* of the IODF and one of the following *file type*s:
>
> **WORKACTL**   for the activity log file of a work IODF
>
> **PRODACTL**   for the activity log file of a production IODF
>
> **BACKACTL**   for the activity log file of a backup IODF
>
> Examples:
> ```
> IODF01 WORKIODF A
> IODF01 PRODIODF A
> ```
>
> For more information on naming conventions for z/VM HCD refer to *z/VM: I/O Configuration* (SC24-6044).

To populate a new config file with data from an existing IODF, see "Creating a new configuration from an IODF" on page 72.

Opening an existing configuration

If you are opening an existing configuration to edit, you can either:
- Open an existing config file and associated IODF, or
- Open an existing config file and an alternate IODF, and
- Upgrade earlier versions of configuration files.

Opening an existing configuration file and associated IODF

Choose the **Open...** command from the **File** menu. Select an existing configuration file to open from the list in the standard **File Open** dialog that appears. Note that HCM appends the file extension *.HCM to your configuration file in case no extension is specified yet. HCM opens the configuration file against the particular IODF, checking to ensure they are in sync.
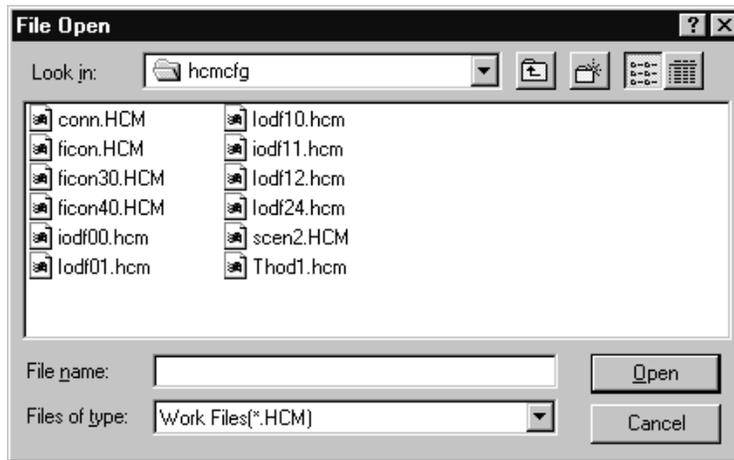
Figure 40. File Open Dialog

If the config file and IODF are out of sync, HCM will ask if you wish to open the configuration file anyway. Either the IODF has been replaced with an earlier or even unrelated IODF, or it has been updated on the host, outside HCM. If you open the config file, HCM will resynchronize; the configuration file will be altered to match the changed IODF.

Opening an existing configuration file and a different IODF

You can deliberately open a configuration file with an IODF other than the one associated with it. For example, you may wish to fix a mistake using an earlier IODF, or want to copy the IODF to a new (alternate) one with a larger allocated size. If the IODF has been renamed on the host, this feature allows you to open the configuration file against the renamed IODF.

Choose **Open Special...** from the **File** menu. The **File Open** dialog is displayed. Select a configuration file and click on **OK**.

Then, when working on z/OS, the **High-level Qualifier** dialog is displayed where you can specify a qualifier which determines the list of IODFs to be displayed. Click on **OK**. When working on z/VM, this dialog is skipped.

The **Select an Alternate IODF** dialog is displayed:
- for **z/OS**, listing the IODFs with the specified qualifier,
- for **z/VM**, listing all IODFs available on your accessed minidisks.

Select an IODF from the list and press **OK**. The configuration file is opened and resynchronized against the selected IODF.

Upgrading configuration files

Using a config file created with an earlier version or service level of HCM can require an upgrade if you try to open this configuration file with a newer version of HCM. HCM applies an upgrade to both work configuration files and production configuration files. With HCM you can:
- upgrade the content of work and production configuration files (older *.HCR and *.HCM files are usable with newer versions or service levels of HCM)
- access the content of work and production configuration files as read-only (the *.HCR and *.HCM files are not upgraded)

HCM automatically detects whether a config file of an earlier version or service level needs to be upgraded for the use with the current level of HCM. The upgrade operation is indicated by the following dialog box:
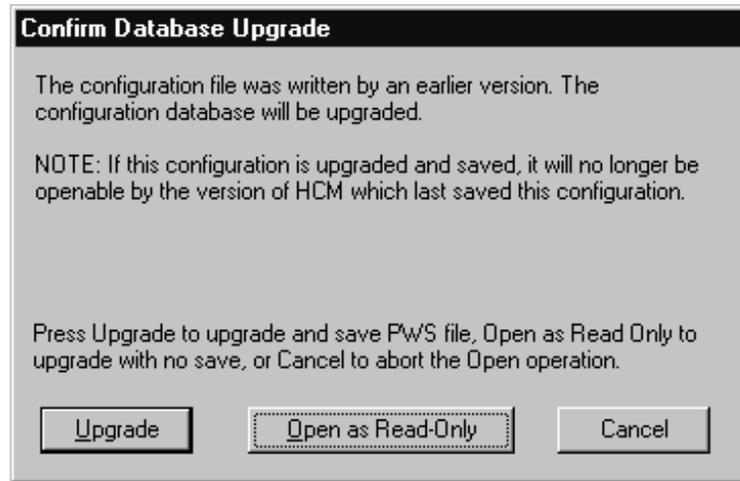
Figure 41. Confirm Database Upgrade

If you select the **Upgrade** button, HCM performs the upgrade operation and then saves the config file.

After the upgrade of the configuration to the new version or service level has been done, a resync of the configuration with the IODF is performed automatically.

If you select the **Open as Read-Only** button, the configuration file is not saved and you are not allowed to edit the config file - regardless whether a work or production configuration file is open.

Note: In case you want to continue to use your configuration file with an older version or service level of HCM, you must create a backup of it, which you can use in conjunction with the older version or service level of HCM.

Working with your IODFs and configuration files

This chapter explains how you can handle your IODF files with HCM.

Copying an existing IODF

Select an IODF from the list and click on the **Copy...** button in the **IODFs** dialog to open the **Target IODF** dialog. If you are copying to a new IODF file, type in the new IODF filename in the text field. If you are copying to an existing IODF file, HCM will confirm overwriting the file.

Making a work or a production IODF

A *work IODF* can be modified and updated; it serves as your "working copy".

A *production IODF* is used by IPL to build UCBs and EDTs. It is also used to build IOCDSs and IOCP input data sets. Several users can view a production IODF concurrently and make reports of it, but it cannot be modified.

If you build a production IODF from a work IODF where XMP processors are defined, a production IODF can only be built if all logical channels (CHPIDs) of all XMP processors have PCHID values defined. If at least one PCHID value is missing, HCM does not build a production IODF, but flags the work IODF as *validated*. In this case, you will receive at least one error message saying that channel path(s) do not have a physical channel assigned, and you also get an informal message saying that the current IODF is now valid except of missing PCHIDs. Select one of the following alternatives to integrate the missing PCHIDs into the work IODF:

- **Alternative 1:** The most comfortable method is to invoke the *CHPID Mapping Tool Support* from either the **Utilities** menu or from the *Edit Processor* dialog (see "Editing a processor" on page 100). This alternative is further described in "How to use the CHPID Mapping Tool Support utility."

- **Alternative 2:** Switch to HCD and use the validated work IODF to generate an IOCP input file suitable for the use with the CHPID Mapping Tool. You can find detailed information on how HCD can interact with the CHPID Mapping Tool in the applicable *z/OS HCD User's Guide* or *z/VM: I/O Configuration* book. The output of this process now is a complete and valid work IODF from which you can build the production IODF. When loading the modified IODF back into HCM again, you need to resynchronize HCD and HCM data as described in "The IODF load / resync process" on page 67.

- **Alternative 3:** You can use the mapping information provided by the CHPID Mapping Tool to type in the PCHIDs manually into the HCM **Create CHPIDs** dialog (see "Create CHPIDs" on page 107).

With the **IODFs...** dialog you can create work and production IODFs: either via the **Build Prod...** or the **Make Work...** action. These actions create a work or a production IODF but they do *not* create the associated work and production configuration files.

Building a work or production IODF via the **IODFs...** menu means, that the respective IODFs are created only (see "Create work and production IODFs without the associated configuration files" on page 86).

Note: We recommend that you should build a production IODF or work IODF together with the associated configuration file (see "Create work and production IODFs with the associated configuration files" on page 87).

How to use the CHPID Mapping Tool Support utility

HCM can prepare an IOCP file as input to the CHPID Mapping Tool (CMT) in order to add or update PCHIDs to this IOCP for XMP processors. You can download the CHPID Mapping Tool from the Internet at the IBM Resource Link™ Web site at:

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

It runs on a Windows workstation. Input to the CHPID Mapping Tool is the hardware configuration file (CFReport [1]) of your machine and a valid IOCP input file (with missing or obsolete PCHIDs).

---

1. When a machine is ordered, the output of the order process is a binary file that represents the physical description of the final machine. One of the components of that file is the type and physical location, including the Physical Channel Identifier (PCHID) value assigned to that location, of all the I/O features in the final machine. This file is called a CFReport.

In HCM, invoke the *CHPID Mapping Tool Support* dialog from either the **Utilities** menu or from the *Edit Processor* dialog (see "Editing a processor" on page 100). This utility offers two functions:

- It exports (downloads) an IOCP file of a selected processor from the host. With this IOCP file as input, you manually invoke the CHPID Mapping Tool which assigns the PCHID values to the CHPIDs in the IOCP file and saves the modified file on the workstation.

- It imports (uploads) the modified IOCP file back to the host where it automatically invokes a PCHID migration. Then it also refreshes the local configuration with the new CHPID-PCHID assignment.
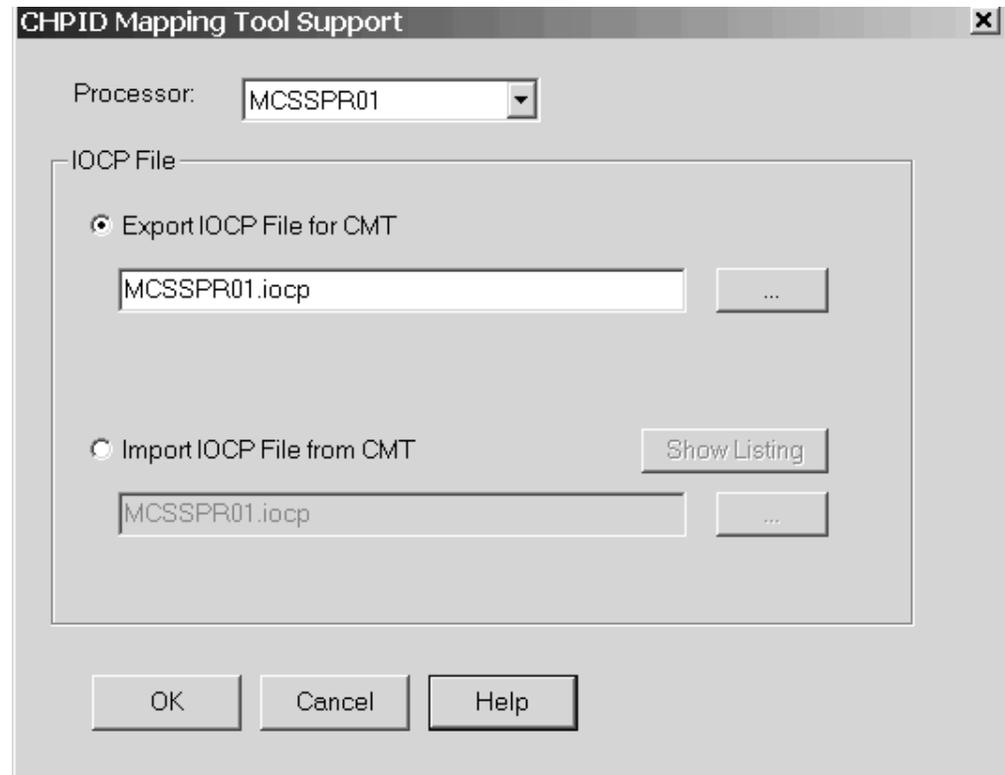
Figure 42. CHPID Mapping Tool Support

Using the dialog shown in Figure 42 you can update PCHID values in your configuration performing the following steps:

1. From the **Processor** listbox, you must select a processor, if you invoked this utility from the **Utilities** menu. If invoked from the *Edit Processor* dialog, this listbox contains the current processor, and you cannot select a different one.

2. Selecting the *Export IOCP File for CMT* radio button receives the IOCP file for the selected processor from the host. The IODF for the current configuration must be a work IODF. The dialog proposes a name for the file where to store the IOCP file on the workstation, or, using the "..." button, you can browse for a file where to store the retrieved IOCP. When you press the *OK* button, HCD creates a validated work IODF on the host and exports an IOCP file for the selected processor. HCM then downloads the exported IOCP file and saves it on the workstation in the previously specified file. If HCD encounters errors during generation of the validated work IODF, HCM displays the corresponding messages and does not download any IOCP file.

3. HCM does not automatically call the CMT. You have to invoke it manually after a successful IOCP file download, providing this file as input. For information on how to use the CMT, refer to its online help and related documentation. The output of a successful CMT run is again an IOCP file which contains the original I/O definitions together with inserted and/or updated PCHID values. The original hardware configuration token is still contained in the generated statements.

4. Select the *Import IOCP File from CMT* radio button and press the *OK* button to send the updated IOCP file back to the host. This action automatically migrates the new CHPID-PCHID assignment into the affected IODF on the host, thus creating an updated work IODF without further need to use the HCD dialog. This action also refreshes the HCM configuration file on the workstation with the PCHID assignment, thus keeping the configurations on host and workstation consistent.

Note: If the PCHID migration produces an assembler error listing, you can use button *Show Listing* to display an assembler listing for error analyzing purposes.

Create work and production IODFs without the associated configuration files

To make a work IODF from a production IODF, click on the **Make Work...** button in the **IODFs** dialog. The **Create IODF** dialog appears, allowing you to create the work IODF. HCM deletes an existing IODF, and creates the new IODF.

To make a production IODF from a work IODF, click on the **Build Prod...** button in the **IODFs** dialog. The **Create Production IODF** dialog appears to allow you to create the production IODF.

When you make a production IODF, a "post-validation" is performed which might issue messages you have to deal with, according to their severity. The production IODF is not created if any errors with a severity higher than 'warning' are produced.

In order to use a production IODF for IPL and dynamic activate, its name must have the format, 'hhhhhhhh.IODFcc' where 'hhhhhhhh' is the high-level qualifier, up to 8 characters long, and 'cc' are any two hexadecimal characters. Optional qualifiers that are allowed for work IODFs must be omitted.

For information on naming conventions for IODFs on z/VM refer to "Creating a new configuration from scratch" on page 79.

Deleting an existing IODF

Select an IODF from the list in the **IODFs** dialog, and click on the **Delete...** button. HCM will display a confirmation dialog prior to deleting the IODF.

Copying your configuration

You can backup the config file and its IODF before making changes to them. Use the **Copy to Backup** command in the **File** menu to open the **Select Target IODF for Copy** dialog. You can either create a new file or select an existing file for the backup. HCM will make copies of the current files, leaving the original files open for you to continue working. If you have changed the IODF or config file, use the **Save** command in the **File** menu to save the changes before using the **Copy to Backup** function.

You can also copy the IODF to just increase the IODF file size.

Create work and production IODFs with the associated configuration files

Building a work or production IODF using the **Build Production...** or the **Make Work...** function of the **File** menu means, that the associated configuration files are also created.

To make a production IODF from a work IODF select **File** from the menu and then select the **Build Production...** action.

Two dialogs appear. The first one is **Create Production IODF** followed by **Save Production File**.

HCM checks whether the target IODF exists on the host. If it already exists, HCM deletes the existing IODF and proceeds with the **Create Production IODF** operation.

The following results may occur when the **Build Production...** action is performed on the host. The first two results indicate that the operation was performed successfully.

SUCCESS	No warnings or errors. The *.HCR file is to be saved under the name supplied in the **Save Production File** dialog.
SUCCESS WITH WARNINGS	
	No errors. You can view the warnings. HCM continues with the **Save Production File** dialog. If you do *not* want to continue, the production IODF is still available. There is *no* automatic deletion of the IODF.
ERROR	No production IODF has been created. You can view the error and warning messages.

If the production IODF is successfully created, HCM displays the **Save Production File** dialog. The *.HCR file is saved under the name supplied in the **Save Production File** dialog.
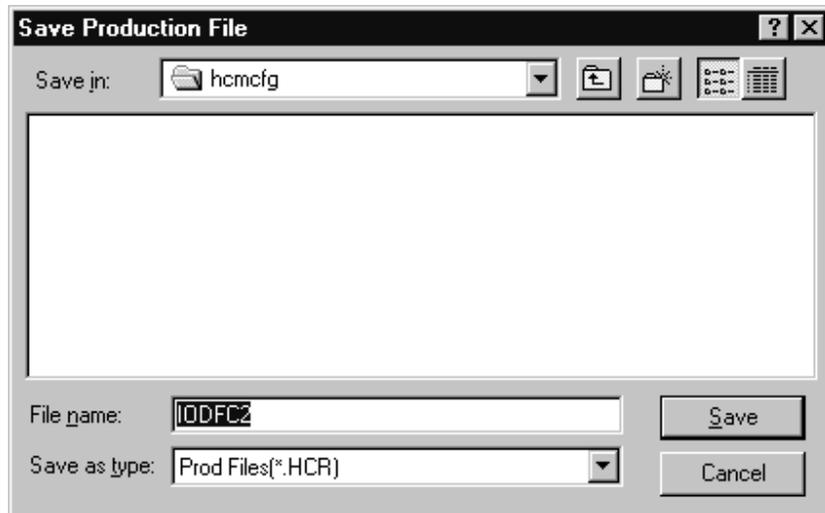
Figure 43. Save Production File Dialog

In the Save Files of Type combination box you can select between Prod Files (*.HCR) and All Files (*.*).

When saving a production configuration file, you can specify the path in the Directories and Drives section, the file name in the File Name section, and the file type in the Save File as Type section. The defaults are:

Path The same path as for the *.HCM file, based on the last used directory.

File Name The same file name as for the *.HCM file name.

File Type *.HCR (If no extension is specified, the default will be used.)

To make a work IODF from a production IODF, select **Make Work...** from the **File** menu. Two dialogs appear. The first one is **Create IODF** followed by **Save Work File**.

You can reach the **Create IODF** dialog from various menus. The following description is for the path via the **Make Work...** action.

HCM checks whether the target IODF exists on the host. If it already exists, HCM deletes the existing IODF and proceeds with the **Create IODF** operation. If the work IODF was successfully created, HCM displays the **Save Work File** dialog.

When saving a work configuration file, you can specify the path in the Directories and Drives section, the file name in the File Name section, and the file type in the Save File as Type section. The default extension is *.HCM.

Naming convention of your configuration files and IODFs

Normally, a given production IODF will repeatedly go through a cycle of being converted into a work IODF so that it may be altered, then converted back into a production IODF for use in the production environment. A standard naming convention for PWS files and IODFs which are part of this cycle can simplify administration. For instance, one possible convention would assign to each production IODF a name that has an ever increasing version number. In an HCM environment, that IODF would be paired with, for example, a IODF92.HCR file on the PWS. When IODF92 requires modification, a work IODF and a work PWS file are generated. The IODF receives the name IODF93.WORK and the PWS file

receives the name IODF93.HCM. Once the changes are made, new production versions are built with the names IODF93 and IODF93.HCR.

Deleting a configuration file

You cannot delete a configuration file together with its associated IODF with one function in HCM. You can delete them separately.

Saving and closing your configuration

The **Save** command in the **File** menu explicitly copies the config file from memory to disk, but keeps it in memory so you can keep working. The **Save** command also saves the IODF, and tags the config file and IODF with matching tokens.

The **Close** command saves both the IODF and the config file, then closes both files. When you close the IODF with **Close** or **Exit**, the **Activity Log** dialog will appear if activity logging is enabled for the IODF.
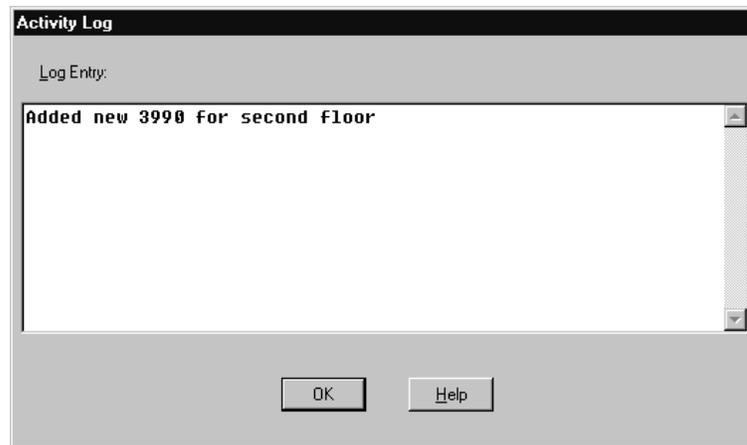
Figure 44. Activity Log Dialog

HCM allows you to add a message to the activity log. For z/OS, use the HCD dialog to view the complete activity log. For z/VM, use BROWSE or XEDIT to view the activity log. See "Creating a new configuration from scratch" on page 79 for this file's naming convention.

You must close an open config file before you can choose **IODFs...**, **Open...**, or **Open Special...** . **Exit** performs a **Save** before exiting.

If HCM terminates because of a host abend, a loss of network connection, or an HCM abend, then the next time you open your config file HCM may need to resynchronize it with the IODF. Furthermore, whether a resync is required or not, the last modification made before the termination may need to be made again.

Chapter 5. Standalone mode

In addition to the HCM online mode of operation where you have a connection to your host, HCM offers the standalone mode. In this mode, you can open a configuration file *without* a host connection. The standalone mode extends the scope of HCM with a view to mobility and problem situations.

In standalone mode you can open both work and production configuration files. The production configuration file contains the complete IODF information, therefore, you can view the data associated with that configuration even if the host is not available. In this case you can use HCM as a browser for the configuration, but you cannot update any data (read-only mode).

If the configuration file is a work configuration file, its physical data (excluding connections) can be edited. This allows you to gather cable information, VOLSERs, user data fields, and other information for example, in the machine room floor while editing the configuration on a portable workstation. In the work configuration file, you can shuffle control units between physical controller segments and move devices from one unit in a string to another unit. This is possible as long as the logical definitions are not affected.

Invoking the standalone mode

To invoke the standalone mode select the **Standalone** button on the **Login** dialog window. As a prerequisite, your configuration file must reside on a resource on the workstation (or accessible from the workstation).
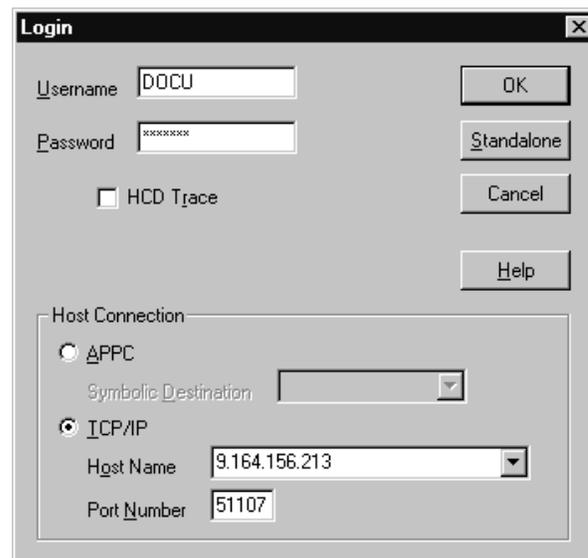
Figure 45. Login Dialog

Working with the standalone mode

The standalone mode disables all actions within a configuration which:
• Require a connection to the host.

- Are related to objects and relations not stored in the work configuration file. Refer to "Work configuration file" on page 93.
- Require a resynchronization between the IODF and the configuration file.

If you work with the standalone mode, the most obvious deviation from the online mode are the following disabled menu choices:

Table 8. Disabled Menu Choices

File	Edit	Create	Options
Open Special... Copy to Backup... Build Production... Make Work...[2] IODFs... Import Data ...[3]	OS Configurations ...[2]	Processor... Switch... Crossbar Switch... Controller... String... CTC Connection... CF Connection Cabinet ...[3]	Activate HCD Trace

The menu bar choices **Operations** and **Utilities** are also not available.

Creating objects

Most operations to create objects in standalone mode are not allowed because they imply the creation of at least one logical IODF object. For instance, HCM does not provide a means of creating an empty controller. At least one CU is always created. Therefore, the creation of empty strings and empty controllers is not supported in standalone mode.

The only creation operation which is allowed is the creation of cabinets. Cabinets are physical objects. Physical objects may be connected and disconnected. The connection and disconnection of logical objects is not allowed.

Viewing objects

You can view detailed information when you select **Edit** from the menu.

Printing reports, diagrams, cable labels

You can print reports, existing diagrams, and cable labels in standalone mode for work and production configuration files. The following table illustrates which kind of report may be printed from which configuration file:

Table 9. Printing of Reports

Type of printing	*.HCM file	*.HCR file
Report	No[4]	Yes
Existing diagrams	Yes	Yes
Cable labels	Yes	Yes

---

2. These actions are disabled when a work configuration file is opened.

3. These actions are disabled when a production configuration file is opened.

4. Report printing requires some data which is available only from the host or from production configuration files.

No host connection

Regardless of whether a work or a production configuration file is open, any action that requires a host connection, produces the message EEQ0212E:

Figure 46. Message EEQ0212E

Work configuration file

If the open file is a work configuration file, HCM is not able to report or display certain data because the data is not present in the work configuration file. This data is only stored in the IODF on the host. In detail the following information is *not* available in standalone mode:
- Switch and port configurations
- Operating system related data
- Logical data of control unit-processor definitions
- Logical data of device-processor definitions

Production configuration file

Within an opened production configuration file you cannot modify the physical and logical data. The production configuration file is read-only.

Note: Some data can only be displayed after selecting the **Edit** button in the appropriate dialogs even if the production configuration file is read-only.

You can use the standalone mode in combination with production configuration files in your enterprise for online documentation of the configuration because you can distribute the files electronically or you can place it on a central LAN resource for common access.

Upgrading

If HCM is running in standalone mode, the upgrade operation is not possible: upgrading requires a host connection.

Resynchronizing

If HCM is running in standalone mode the resynchronization operation is not possible.

VOLSER reconciliation

Before the introduction of OS/390 1.3.0 HCD, VOLSER data could be stored in HCM configuration files, but not in the IODFs. Since the introduction of OS/390 1.3.0 HCD, however, IODFs now contain VOLSER data, and there might be a mismatch between the VOLSER data in an HCM configuration file and its corresponding IODF on the z/OS host if you have updated VOLSER data in the HCM configuration in standalone mode.

The next time you open your configuration file in HCM in online mode by using **Open...** or **Open Special...** from the **File** menu, differences between the VOLSERs stored in your HCM configuration file and those stored in the IODF will be detected. These differences must be unified, or reconciled, and so HCM will present the **VOLSER Reconciliation** dialog to resolve the differences before displaying the configuration diagram.
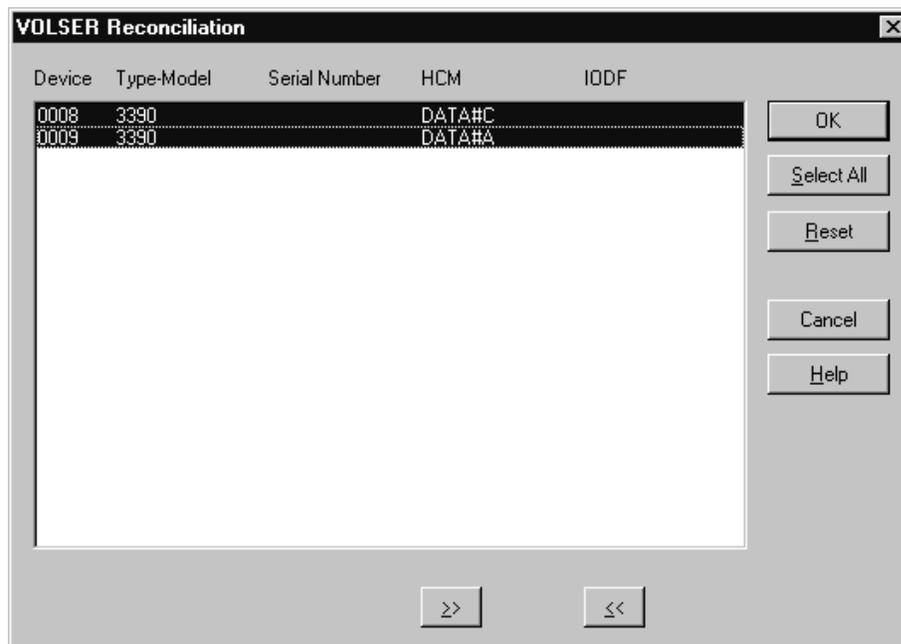
Figure 47. VOLSER Reconciliation Dialog

The listbox contains one line for every device in the configuration for which the VOLSER value in the IODF on the z/OS host and the associated configuration file on the PWS differ. On entry, all lines are preselected. But you can also work with only one or several devices by clicking on the appropriate lines.

Each line of the listbox shows the device ID, the device type and model, the serial number, and the VOLSER values for the configuration file and the current VOLSER recorded in the IODF on the host.

You can process the selected lines in the listbox as follows:
* Click the button with the >> symbol to accept the HCM VOLSER value for the selected device. This button symbolically sets the direction of the VOLSER transfer for the selected lines and copies the HCM VOLSER values into the IODF.

- Click the button with the << symbol to reject the HCM VOLSER value for the selected device. This button symbolically sets the direction of the VOLSER transfer for the selected lines and copies the IODF VOLSER values into the HCM configuration file.

Once a line has been processed, the selected lines are grayed and a >> or << symbol is inserted between the HCM VOLSER value and the IODF VOLSER value.

To unprocess your selection you can use the **Reset** button to mark all the lines in the listbox as requiring processing again. The lines can now be reprocessed again.

When you have processed every item in your listbox, click on the **OK** button. HCM then updates the local and/or the remote database with the accepted VOLSER changes.

Note: HCM offers a "Prime the IODF with Sensed Data" utility in the Utilities menu. Using this utility, sensed data can be stored in the IODF. See "Priming the IODF with sensed data" on page 204 for details.

Chapter 6. Modifying objects in your configuration

Once you have built your initial configuration, you can modify objects in the HCM config file. This section explains how to manipulate all the elements of your hardware configuration with the interactive configuration diagram and dialogs. After you have edited objects in the diagram, you can connect (or disconnect) them, see Chapter 7, "Connecting objects in your configuration," on page 155.

Note: Status information is available only with I/O Operations for z/OS.

Processors

> **A note on terminology:**
> Throughout this document, the following terms are used:
>
> **XMP processor and SMP processor**
> > The term **XMP processor** designates processors that support multiple logical channel subsystems. It is used in contrast to the term **SMP processor**, which designates processors of previous generations that support only one channel subsystem.
> >
> > For XMP processors, HCM offers methods to explicitly define multiple logical channel subsystems (LCSS). For SMP processors, the single channel subsystem is implicitly defined with the processor.
>
> Wherever a channel subsystem of a XMP processor is referenced in dialogs, you will see the channel subsystem ID (CSS ID) appended to the corresponding processor ID, like `procname.n` where `n` is the CSS ID.

You can create, edit or delete processors by invoking the appropriate dialogs for the desired processor object in the configuration diagram. Double-clicking on a processor object displays the **Processor** dialog where you can define partitions and CHPIDs, or for XMP processors, you can also define channel subsystems. Alternatively, choose **Processor...** from the **Edit** menu and select the processor name from the list that appears. Note that you can rearrange and filter the list according to your needs.

The **Processor** dialog displays detailed information for the selected processor and the associated system. It allows you to navigate to other related dialogs. You can add or edit partitions, CHPIDs, the processor details or check the system status (only active when online and in BASIC mode and when I/O Operations is up and running) by pressing the appropriate button.

You can also change a processor's style, make it a primary or alternate processor, or assign processors to machines during IODF load / resync. See Appendix B, "Alternate processor definitions," on page 407.

There are two versions of the **Processor** dialog, one for SMP processors and one for XMP processors (Figure 48 on page 98 and Figure 49 on page 98). The one for XMP processors contains a **Channel Subsystems** section which lists the channel subsystems (CSS) defined for the processor. To edit partitions and channels for an

XMP processor, you must select a CSS first, as these objects are pertaining to a specific CSS.
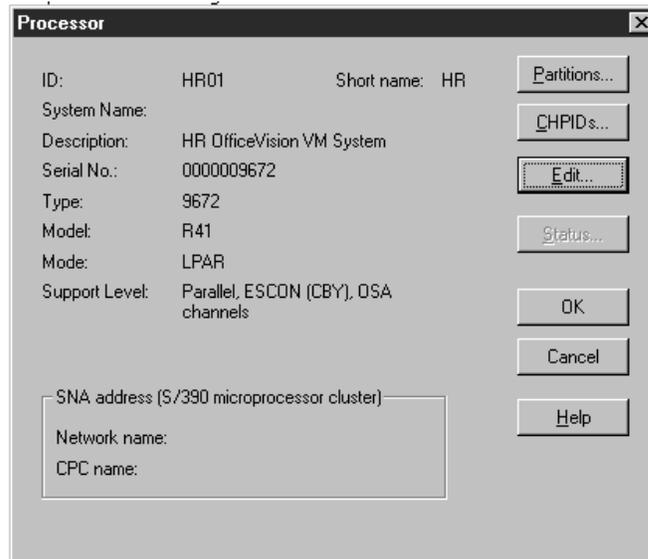
Figure 48. Processor Dialog for SMP Processors
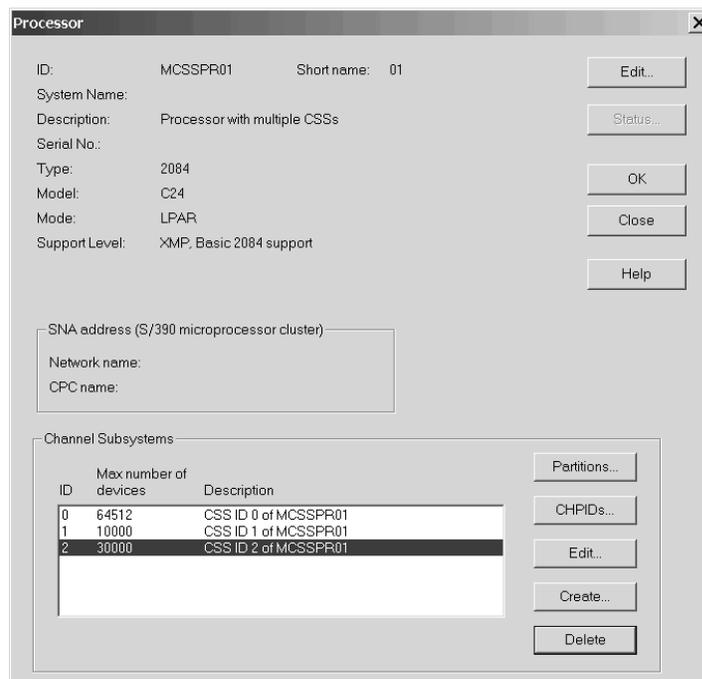
Figure 49. Processor Dialog for XMP Processors

Creating a processor

Display the **Create Processor** dialog by choosing **Processor...** from the **Create** menu. Type in the fields or select data from the drop down boxes. The **ID** and **Type-Model** fields are the only required ones. Note that **Type-Model** is a single field; you can select a type and model from the drop down box or type in the field.

For XMP processors, in the field **Number of Channel Subsystems** you may specify the number of channel subsystems to be defined for your current processor. For XMP processors, the default value is 1. This will create a channel subsystem with ID 0. A value of 2 will create CSS 0 and CSS 1 and so on. The value is ignored for SMP processors. Also, the configuration mode defaults to LPAR, which is the only allowed mode for XMP processors.

After you have filled in the information, press **OK** to create the new processor, which will appear in the configuration diagram. You can then double-click on the processor to display the **Processor** dialog (see Figure 48 on page 98).

Note that the **Short name** of a processor is part of the physical machine.
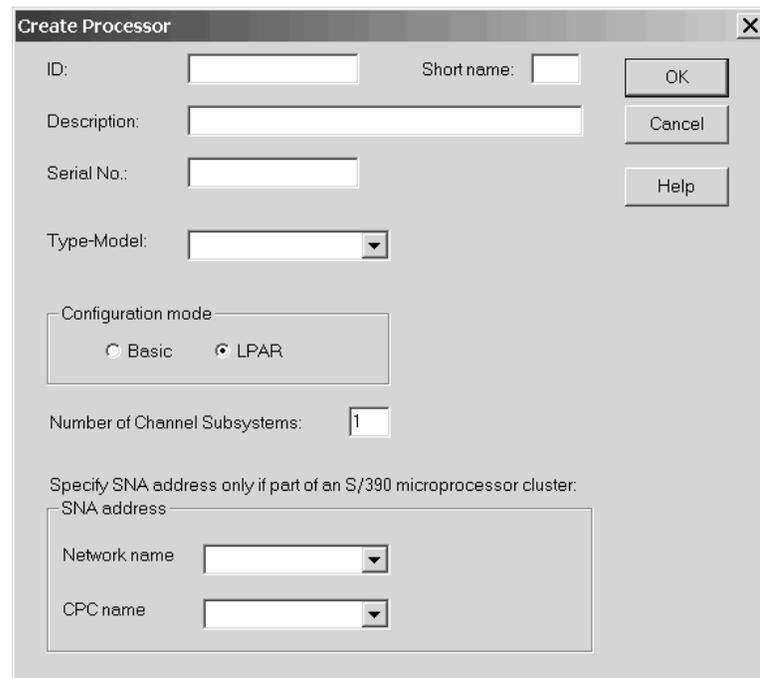
Figure 50. Create Processor Dialog

> **Tip:**
> To create another processor for "one processor per partition" processors, use the **Create...** button in the **Processors** dialog. (Display the **Processors** dialog by choosing **Manage Shadow Processors...** from the **Edit** menu.) See "Managing shadow processors during an HCM session" on page 407 for details.

Choosing a support level

If there is more than one support level for the type and model of processor that you specify in the **Create Processor** or **Edit Processor** dialog, the **Choose Support Level** dialog appears when you press **OK**. Select the support level for the selected processor and press **OK**. Selecting the correct support level is important for validation processing, as the support level controls the rules that apply to the processor (for example, the number of CHPIDs).

Editing a processor

Click on the **Edit...** button in the **Processor** dialog to display the **Edit Processor** dialog. You can then add or edit processor information and press **OK**. The changes will appear in the **Processor** dialog's list box. If there is more than one support level for the type and model of processor specified, the **Choose Support Level** dialog appears with the current support level highlighted when you press **OK**.

If you change the type, model or support level of the processor, you may have to renumber the CHPIDs. For example, if you change the processor type from one with 64 CHPIDs to one with 128, the CHPIDs which were originally divided equally between the two sides of the processor (32 on side A and 32 on side B) will now appear all on one side (64 on side A). You may wish to renumber the CHPIDs such that the original 64 CHPIDs plus the extra new 64 CHPIDs are divided equally between the two sides. HCM will propose new CHPID numbers as appropriate for the new CHPIDs.

If you have changed the Type/Model or Support Level for an SMP processor, pressing **OK** in the **Edit Processor** dialog will display the **CHPID Renumbering** dialog, allowing you to renumber the CHPIDs as required.

The **Edit Processor** dialog contains a **CMT...** button. This button invokes the *CHPID Mapping Tool Support* dialog. Use this dialog to interact with the CHPID Mapping Tool to assign missing PCHIDs to CHPIDs (or update existing PCHID-CHPID assignments) for XMP processors. See "How to use the CHPID Mapping Tool Support utility" on page 84 for further information.

Adding user data to a processor

Add or edit user data by pressing the **Info...** button in the **Edit Processor** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Deleting a processor

Select the processor object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the processor.

You can also delete processors in the **Processors** dialog, called by choosing **Manage Shadow Processors...** from the **Edit** menu. Select a logical processor in a machine and press the **Delete** button. HCM will display a confirmation dialog, then delete the processor, provided that it is not the last one within the machine. The primary and alternate settings are adjusted for the remaining processors in the machine. See Appendix B, "Alternate processor definitions," on page 407.

Channel subsystems

> **Note:**
> All tasks described in this chapter's subsections are only available for XMP processors.

Once you have created an XMP processor, you can add and modify its channel subsystems. First select an XMP processor by double-clicking on the processor object in the configuration diagram; alternatively, choose **Processor...** from the **Edit**

menu and select the processor name from the list that appears. In the upcoming **Processor** dialog, you see the **Channel Subsystems** section which lists the channel subsystems defined for this processor.

Creating or editing a channel subsystem

Invoke the **Create Channel Subsystem** dialog from the **Create...** button in the Channel Subsystems section of the **Processor** dialog. It contains three controls: **ID**, **Maximum number of devices** and **Description**. The **ID** field is initialized with the first unused CSS number.
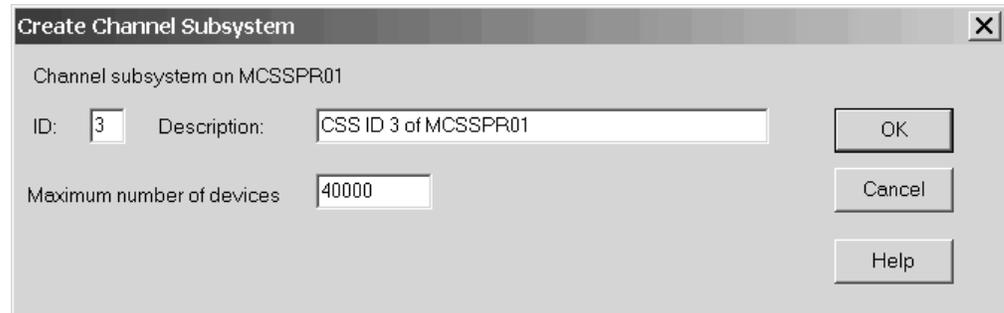
Figure 51. Create Channel Subsystem Dialog

If you want to edit an existing CSS, press the **Edit...** button in the Channel Subsystems section. The upcoming **Edit Channel Subsystem** dialog lets you change the number of devices and the description for the selected CSS.

> **Note:**
> A channel subsystem becomes visible in the configuration diagram only after you have created its first partition. From then on, a channel subsystem's pop-up menu offers you choices to edit the channel subsystem or to work with its CHPIDs and partitions.

Deleting a channel subsystem

Select a channel subsystem from the list box in the Channel Subsystem section in the **Processor** dialog and click on the **Delete** button. HCM will display a confirmation dialog before deleting the channel subsystem.

Partitions

Once you have created a processor and a channel subsystem for an XMP processor, you can add and modify its partitions. First select a processor by double-clicking on the processor object in the configuration diagram; alternatively, choose **Processor...** from the **Edit** menu and select the processor name from the list that appears. Display the **Partitions** dialog for the processor by clicking on the **Partitions...** button in the **Processor** dialog. For XMP processors, this button is in the Channel Subsystems section.

The list box in the **Partitions** dialog lists all partitions for the selected processor, or for XMP processors, the list of partitions includes only partitions in the selected CSS.

You can create a new partition, or select a partition to edit or delete it.
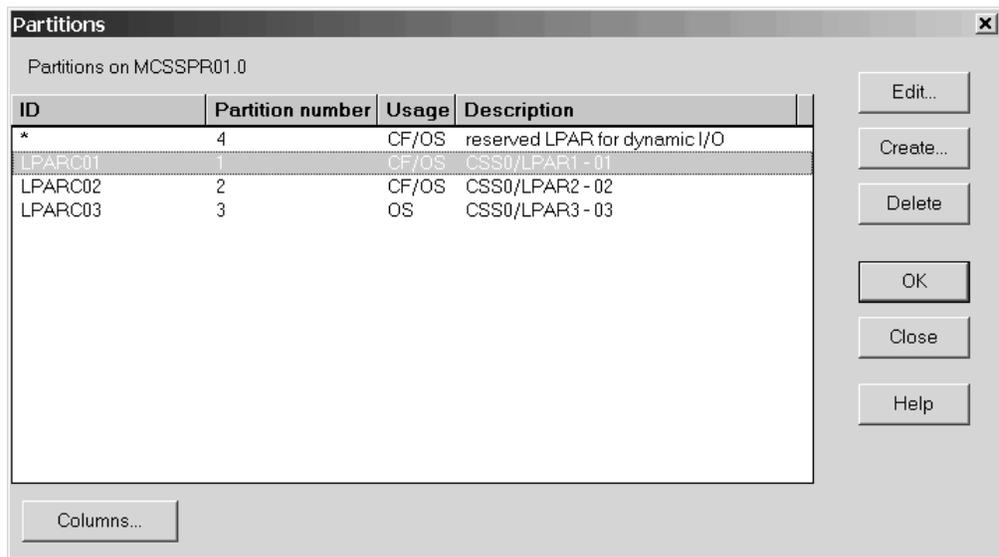
Figure 52. Partitions Dialog

Creating partitions

You can create a new partition for the selected processor. Click on the **Create...** button in the **Partitions** dialog to display the **Create Partition** dialog. Enter the information defining the partition and press **OK**. The partition will appear in the **Partitions** dialog list box. It will also appear as an object in the configuration diagram within the processor or within a processor's channel subsystem where it has been created. Note that the Usage option determines if the partition will be used solely for standard operating system connections (OS), as a coupling facility (CF), or both (CF/OS).

After creating one or more partitions, they can be assigned to the access and candidate lists of CHPIDs. See "Assigning CHPIDs to partitions" on page 110 for more information.

Partitions for dynamic I/O configuration

On XMP processors with the appropriate support level, you can add or remove partitions via dynamic I/O configuration. HCM supports this functionality: with an '*' as placeholder ID, you can define reserved partitions, which you plan to add dynamically to a configuration at a later point in time. With this '*', they appear in the **Partitions** dialog of a channel subsystem and in the configuration diagram. Dynamic partitions do not appear in the **Assign Partitions** dialog of a CHPID, so that you cannot attach any CHPIDs or devices to such partitions.

You can only activate a reserved partition dynamically, if this partition had been included in the initial POR. To activate a reserved partition, you need to edit the partition by changing the '*' name to a valid partition name before building a new production IODF.

Granting I/O devices access to the new partition

If there are already I/O devices connected to the processor containing the new partition, pressing **OK** in the **Create Partitions** dialog will display the **I/O Devices with Explicit Candidate Lists** dialog. You can select the I/O devices which should be granted access to the new partition before HCM creates the partition.

Note that you can display serial numbers and VOLSERs, device descriptions, or control unit numbers in the last column of the device list box by clicking one of the option buttons in the View section.

Editing a partition

You can edit a partition in different ways:

- In the **Partitions** dialog select a partition and click on the **Edit...** button to open the **Edit Partition** dialog for the selected partition.
- Double-click on the partition object within the processor in the configuration diagram.
- Choose **Partition...** from the **Edit** menu and select the partition name from the list that appears. Note that you can rearrange and filter the list according to your needs.

Edit the partition data in the fields and press **OK**.

The **Status** button allows you to retrieve information about this partition and to check system status.

Adding user data to a partition

Add or edit user data by pressing the **Info...** button in the **Edit Partition** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Deleting a partition

Select a partition from the list box in the **Partitions** dialog and click on the **Delete** button. HCM will display a confirmation dialog before deleting the partition. Note that you cannot delete a partition directly from the configuration diagram.

CHPIDs

Channel path IDs or CHPIDs can be added or modified once you have created a processor or a channel subsystem within an XMP processor. First display the **Processor** dialog by double-clicking on the processor in the configuration diagram, or by choosing **Processor...** from the **Edit** menu and selecting a processor from the list box that appears. Then click on the **CHPIDs...** button in the **Processor** dialog to display the **CHPIDs** dialog. For XMP processors, this button is in the Channel Subsystems section of the **Processor** dialog.

Note:

Wherever a CHPID for an XMP processor needs to be identified, it is named in the format `procname.n.xx` where:

- `procname` is the name of the XMP processor
- `n` is the CSS ID within the processor
- `xx` is the CHPID within the CSS (two hexadecimal digits).

Example:

`MCSSPR01.1.0B`

Also, for CHPIDs in XMP processors, you can make the physical channel information visible by adding column **PCHID** into the window.
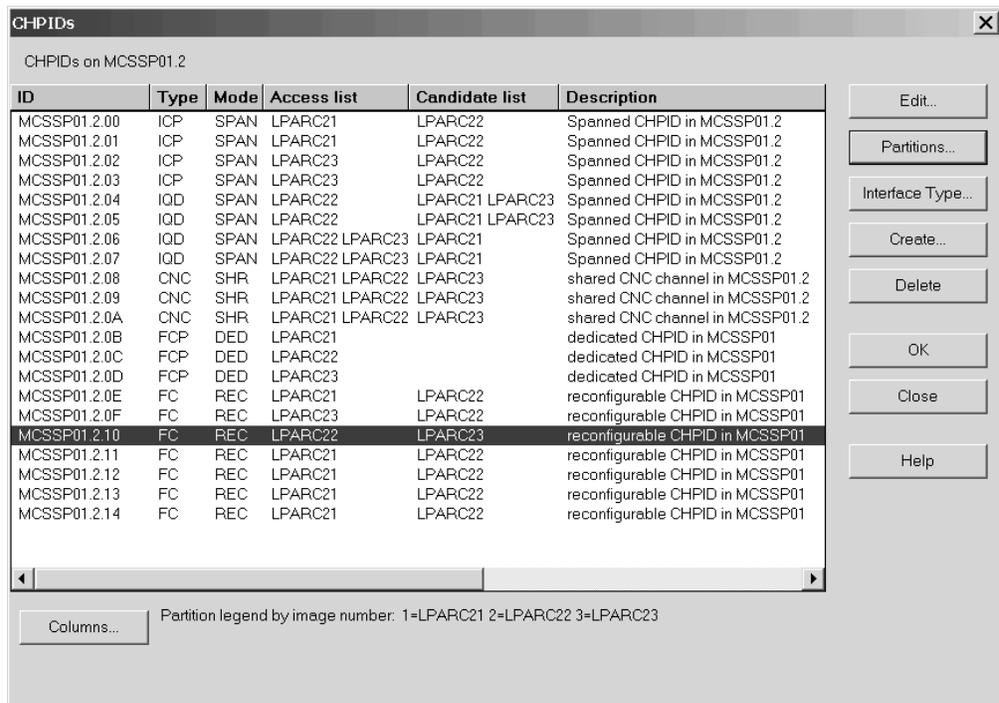
Figure 53. CHPIDs Dialog

The list box in the **CHPIDs** dialog shows all CHPIDs for the selected processor or the selected channel subsystem of an XMP processor. You can create or delete CHPIDs from this dialog. You can select one or more CHPIDs to view details or assign them to partitions.

There may be four special cases of CHPID text displayed in the list box, for example:

ID	Interface type	Type	Mode	
AC	S	CNC	SHR	*Normal case*
AC*		CNC	SHR	*CHPID without a physical channel*
AC*	S	–	–	*Physical channel without a CHPID (columns Type and Mode are blank)*
AC/32	S	CNC	SHR	*A CHPID mapped to Channel 32 (only appears on "one processor per partition" machines)*

CHPID interface type

The (physical) interface type is independent of the (logical) CHPID type, and does not always match (physical-logical inconsistency). To correct inconsistencies, either change the CHPID type in the **Edit CHPID** dialog (press the **Edit...** button in the **CHPIDs** dialog), or change the interface type in the **Change Path Interface Type** dialog (press the Interface **Type...** button in the **CHPIDs** dialog).

Crossbar switch interfaces are always parallel. ESCON or FICON switch interfaces (ports) are always serial. CHPIDs can be any of the five path interface types. Controller channel interfaces can be parallel, serial, IOC or OSA. Note that

converters can only convert between parallel and serial; interfaces of the type coupling, IOC or OSA can only connect to each other.

Changing between serial and parallel interface types can usually be done while the interface is connected; converters will be inserted or removed as appropriate.

CHPID operation mode

The operation mode determines how CHPIDs are assigned to partitions. A CHPID can be dedicated (DED), reconfigurable (REC), shared (SHR), or spanned (SPAN). A *dedicated* CHPID is assigned to one partition; only that partition can access I/O devices on this CHPID. A *reconfigurable* CHPID is an unshared CHPID that you can reconfigure off-line from one partition, then on-line to another (for example, after a power-on reset). Only one partition can access I/O devices on this CHPID at a time. A CHPID that can dynamically access more than one partition is a *shared* CHPID; one or more partitions can access I/O devices using this CHPID.

With XMP processors, it is possible to access I/O devices using one CHPID that is accessed by partitions from multiple logical channel subsystems. Such a CHPID is called a *spanned* CHPID.

More about spanned channel paths in multiple LCSSs

Depending on the processor type, in the HCD dialog you may define certain channel paths with operation mode SPAN, with the result that - depending on what is defined - you are offered partitions from multiple logical channel subsystems in the access and candidate list.

A spanned channel path will be created with the same CHPID number in all channel subsystems that are using it. For example, if you have a processor MCSSPRO1 with channel subsystems 0 through 3, and you create CHPID 1A (type IQD, SPAN) and let it access partitions from CSS 0, 2, and 3, you will get this CHPID 1A in CSSs 0, 2, and 3, but not in CSS 1.

If you define a channel as SPAN, but connect it to partitions from a single channel subsystem only, then HCM displays its operation mode as SHR. The other way round, if a shared channel path accessing partitions from one channel subsystem only, is eligible for being spanned over multiple logical channel subsystems, and you enlarge its access or candidate list with partitions from multiple logical channel subsystems, then HCM displays this channel's operation mode as SPAN.

A channel subsystem that is not using a spanned channel can use the CHPID of that spanned channel for a separate channel path definition.

> **Note:**
> It is dependent on the processor support level which channel path types can be defined as spanned. Managed channels cannot be defined as spanned.

CHPID operation mode symbols

On the configuration diagram, access from a partition to a CHPID is represented by a symbol at the intersection between the CHPID's column and the partition's row. The symbols represent the CHPID's mode (DED, REC, SHR, or SPAN) and whether the partition is in the CHPID's access list or its candidate list.

Cand	Access	
	■	Dedicated (DED)
□	◨	Reconfigurable (REC)
○	◉	Shared (SHR)
◇	◈	Spanned (SPAN)

Figure 54. CHPID Operation Mode Symbols

Unmanaged/managed CHPIDs

HCM supports the definition and visualization of manageable channel paths. Dynamic Channel Path Management is a capability designed to dynamically adjust the channel configuration in response to shifting workload patterns. It automatically adds and removes I/O paths to disk control units in an attempt to provide the most effective resource utilization from the available hardware and configuration. It also coordinates these changes across the systems that share the paths eligible to be reconfigured.

When you define a CHPID as managed, you have to click the **Managed** checkbox on the **Create CHPIDs** dialog and specify an I/O Cluster when the processor is running in LPAR mode.

Selecting a CHPID to view details

Select one or more CHPIDs in the **CHPIDs** dialog and press the **Edit...** button. Alternatively, double-click on a CHPID in the configuration diagram. The **CHPID** dialog will open, displaying the details for the selected CHPID, including ID, description, type, mode, and to what objects the CHPID is connected. If you have selected more than one CHPID, pressing **OK** will cycle through each CHPID's details. You can then edit the CHPID or assign it to one or more partitions.

The **Status** button leads you to the **CHPID status** dialog.
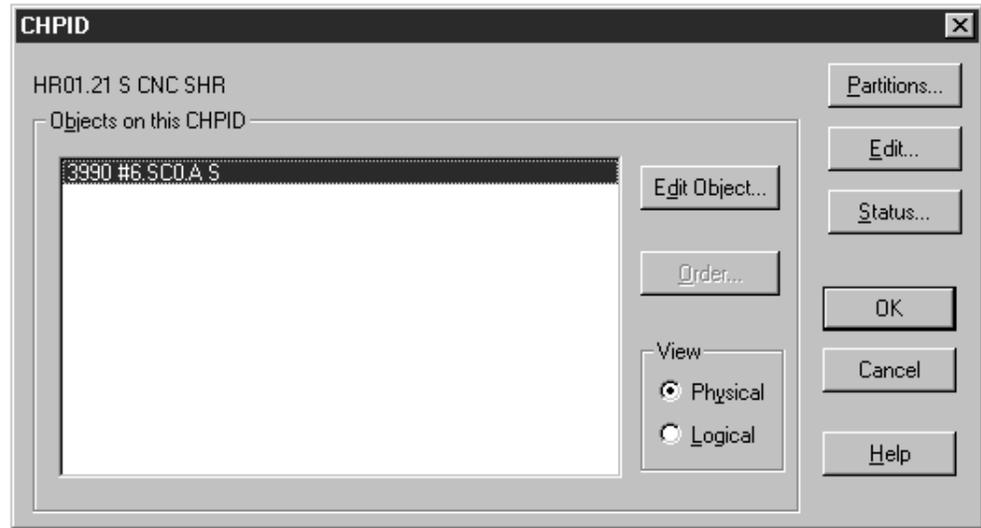
Figure 55. CHPID Dialog

Editing objects connected to a CHPID

The **CHPID** dialog lists the objects connected to a selected CHPID. If you choose the Physical option in the View section, select an object, and press the **Edit Object...** button, the **Edit** dialog for that physical object will open. That is, select a controller to open the **Controller** dialog; select a crossbar switch to open the **Edit Crossbar Switch** dialog; or select a switch to open the **Switch** dialog.

If you choose the Logical option in the View section, select an object, and press the **Edit Object...** button, the **Edit** dialog for that logical object will open. Select a control unit to open the **Control Unit List** dialog, or select a CTC connection to open the **CTC Connection** dialog.

Ordering objects connected to a CHPID

From the **CHPID** dialog, press the **Order...** button. This opens an order dialog, allowing you to change the order in which objects are daisy-chained on this CHPID.

Create CHPIDs

You can create new CHPIDs for a selected processor or channel subsystem. Click on the **Create...** button in the **CHPIDs** dialog to open the **Create CHPID** dialog and enter the CHPID information. For XMP processors, the **CHPIDs** dialog is working in the context of the previously selected CSS.
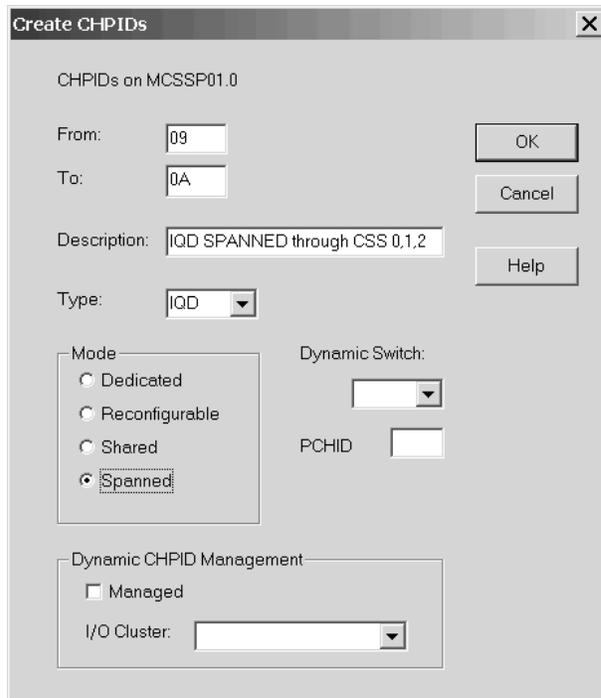
Figure 56. Create CHPIDs Dialog

You typically create a range of CHPIDs with the **From** and **To** fields. For XMP processors, the channel's physical channel identifier (PCHID) is required for each CHPID. If you know this value, you can enter it into the **PCHID** field. If you create a range of CHPIDs, the PCHIDs will be assigned sequentially, starting with the PCHID you entered. If you do not know the channel's PCHID, or the generated values do not match the real values, you need to update them. You can use the CHPID Mapping Tool to establish the PCHID-to-CHPID relation. You may then either enter the PCHID values manually or you may use the process for inserting PCHIDs into a validated work IODF introduced in "Making a work or a production IODF" on page 83 and fully described in the *z/OS HCD User's Guide*. For more information on the CHPID Mapping Tool refer to the applicable *z/OS HCD User's Guide* or *z/VM: I/O Configuration* book.

If a switch is already defined in the configuration, you can indicate the dynamic switch in the **Create CHPID** dialog. Once you have created CHPIDs, they will appear in the **CHPIDs** dialog list box and as objects within the selected processor or channel subsystem in the configuration diagram.

If you create an OSD CHPID that supports queue prioritizing, HCM displays the *Enable / Disable Queue Prioritizing* dialog (Figure 57 on page 109). Select the *Disable Queue Prioritization* checkbox if you want to disable queue prioritizing, which otherwise will be enabled.
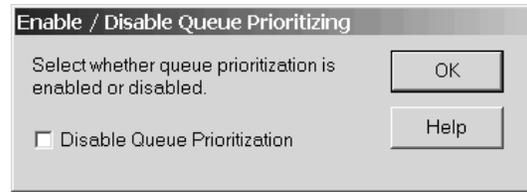
Figure 57. Enable / Disable Queue Prioritizing

Over-defining a CHPID

For an XMP processor, you can define a channel path that is not physically installed on the machine. This may be useful if you want to migrate from a machine which had more channels defined than the target XMP processor has currently installed, or if you want to prepare a configuration for future upgrades of the channel cards.

To distinguish an over-defined CHPID from a physically installed CHPID, use character '*' for the PCHID value when creating the CHPID. An over-defined CHPID must adhere to all validation rules. Over-defined channel paths are not recognized by an IOCDS download, by an IOCP build and by a dynamic activation of an I/O configuration. Also, if a control unit contains only CHPIDs with a PCHID value '*', the whole control unit (including any attached devices) is omitted from the configuration to be activated.

When installing the channel path later, you must edit the CHPID and replace the '*' by its valid PCHID.

If you want to use the CHPID Mapping Tool to assign the correct PCHID, you first need to edit the CHPID and remove the '*'.

Editing a CHPID

You can edit a CHPID's details in different ways:

- Select one or more CHPIDs and press the **Edit...** button in the **CHPIDs** dialog to open the **CHPID** dialog. If you have selected more than one CHPID, you can cycle through each CHPID's details by pressing **OK**, until you find the one you wish to edit.
- Double-click on the desired CHPID in the configuration diagram.
- Choose **CHPID...** from the **Edit** menu and select the CHPID name from the list that appears. Note that you can rearrange and filter the list according to your needs.

In the following **CHPID** dialog press the **Edit...** button to display the **Edit CHPID** dialog.

Note that changing the CHPID type (logical) in this dialog does not automatically change the interface type (physical). If you change the CHPID type such that the interface type is no longer consistent, you must update the interface type with the **Change Path Interface Type** dialog to avoid a physical-logical inconsistency (see "CHPID interface type" on page 104 for more details). Note that you can change the CHPID type while the CHPID is connected.

When you change the CHPID type from one a type corresponding to a parallel interface (BL, BY) to a type corresponding to a serial interface (CNC, CTC, CVC,

CBY), or if you change the operation mode to or from shared (SHR), you must change all other CHPIDs on the same processor that are connected to common control units. In this situation, the **CHPIDs To Be Updated** dialog will appear to allow you to change these CHPIDs at the same time.

For a CHPID that is connected to a controller through a switch, this dialog allows you to enter a two digit hex number for the connected switch. The Dynamic Switch drop down box only lists those switches that are physically reachable, either directly or via one chained switch. If the CHPID is connected directly to a controller, this list is empty. See "Connecting switches to processors or other switches" on page 172 for more details.

Note: For OSD CHPIDs, the **Edit CHPID** dialog contains a checkbox with which you can enable/disable queue prioritization. You can include a column *QP disabled* into the **CHPIDs** dialog which shows a 'Yes' for OSD channels if queue prioritization is disabled, 'No' otherwise.

Deleting a CHPID

Select one or more CHPIDs from the list box in the **CHPIDs** dialog and click on the **Delete** button. HCM will display a confirmation dialog before deleting the CHPID(s). Note that you cannot delete a CHPID directly from the configuration diagram.

If you delete a spanned CHPID, it will be deleted in all channel subsystems where it is defined.

For spanned CHPIDs, removing all partitions of the access and candidate list from a specific channel subsystem (using the **Assign Partitions** dialog shown in Figure 58 on page 111) causes that the CHPID of that particular channel subsystem is removed.

Assigning CHPIDs to partitions

Using the **Assign Partitions** dialog, you assign CHPIDs to partitions in a processor by adding each partition to the CHPID's Access or Candidate list. Each CHPID has two partition lists, Access and Candidate: if a CHPID is either shared or reconfigurable, you can specify which partitions have access to that CHPID, and which are only "candidates" for access. Use the CHPID Access list with or without the CHPID Candidate list.

If you assign partitions to a dedicated or reconfigurable CHPID on an XMP processor, the **Assign Partitions** dialog will be filled with the partitions of the corresponding channel subsystem where the partitions are shown in the format `CSSID.PARTID`, where `CSSID` is the channel subsystem ID and `PARTID` is the partition name. For shared and spanned CHPIDs, the dialog is filled with the partitions of all defined channel subsystems.

To display the **Assign Partitions** dialog, select one or more CHPIDs in the **CHPIDs** dialog and press the **Partitions...** button. Only CHPIDs with the same initial Mode, and Access and Candidate lists may be selected at the same time. Alternatively, press the **Partitions...** button in the **CHPID** dialog. Use the **Add>>** and **<<Remove** buttons to assign partitions to the CHPID Access or Candidate list.
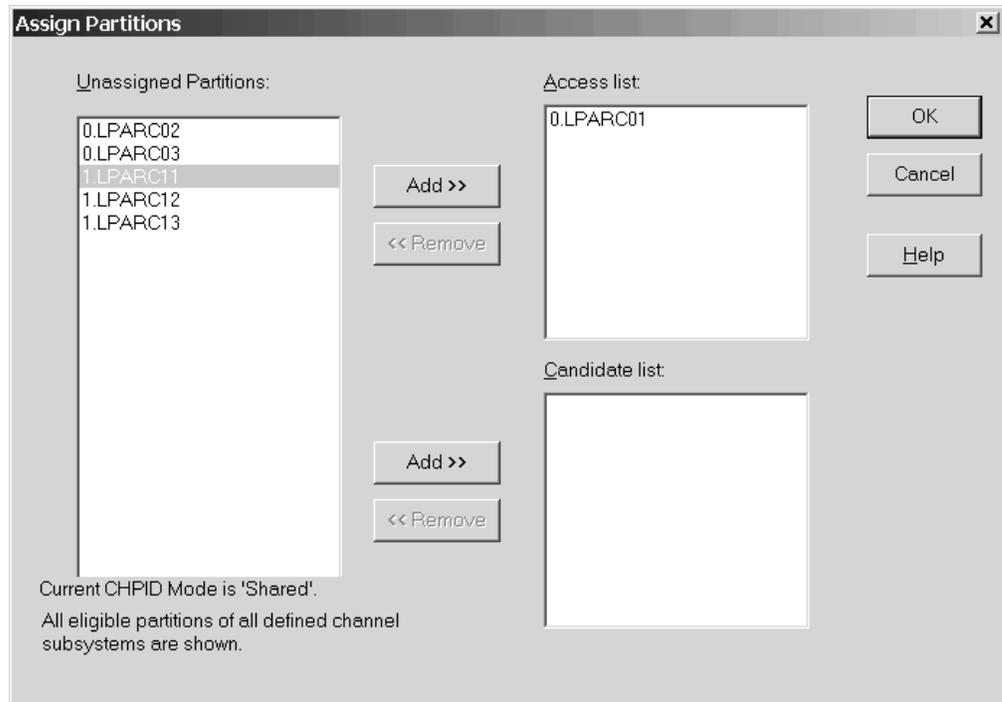
Figure 58. Assign Partitions Dialog

If you assign partitions from multiple channel subsystems to a shared CHPID, then its mode is changed from 'shared' to 'spanned' if the channel can be spanned. Otherwise, if the channel cannot be spanned, you get a message saying that you cannot assign partitions from multiple channel subsystems. Also, if you remove partitions from a spanned CHPID's access or candidate lists, such that it can access only partitions from one channel subsystem, its mode is changed from 'spanned' to 'shared'. A message lets you decide whether you want to accept these changes.

In our example from Figure 58, we are currently assigning partitions to a shared CHPID (see the text below the **Unassigned Partitions** box). Partitions from two different channel subsystems (0 and 1) are available. You see that one partition (LPARC01) from CSS 0 is already assigned. A second partition (LPARC11) from a different channel subsystem (CSS 1) is selected to be added to the Access list. Pressing **Add>>** and then **OK** will display a message that the channel's mode is changed from 'shared' to 'spanned'.

Support of spanned CHPIDs

For your ease of use, a special treatment is applied to spanned CHPIDs in the configuration diagram and in some dialogs.

Physical and logical aspects of spanned CHPIDs

When working with physical CHPIDs (in contrast to internal CHPIDs like IQD or ICP), the difference between their physical and logical aspect might be of interest. Considering the physical view, a spanned physical CHPID has exactly one external interface on a channel card, namely that one into which you can plug a cable. From a logical point of view, a spanned CHPID has two or more instances in different channel subsystems. These can also reach the single external interface on the channel card.

HCM takes care about the differences in the physical and logical aspect as follows:

- Even though the configuration diagram usually only shows the **physical** view of the configuration, it shows all **logical** instances of a spanned CHPID. This allows you, together with HCM's highlighting features, to easily determine over which channel subsystems a CHPID is spanned.
- If you want to physically connect an interface to a spanned CHPID, you can only select that logical instance which is defined to the leftmost channel subsystem.
- If you want to logically connect an interface to a spanned CHPID, you can select all logical instances.

Visualizing spanned CHPIDs

Whenever you include one instance of a spanned CHPID into a view of the configuration diagram, either by filtering or locating options, then all instances of the spanned CHPID are also made visible. Also, if the processor to which the spanned CHPID belongs, is not contained in the view, HCM adds the processor to the view. If you select one instance of a spanned CHPID, then all instances are highlighted. Information about the spanned CHPID is displayed in the status bar.

In our example in Figure 59, you see the selected spanned CHPID 18 of processor MCSSPRO01 with PCHID 31D. This CHPID is spanned over channel subsystems 0 and 1 ([0/1]), so it is highlighted in both channel subsystems. The instance from CSS 0 is connected to port 06 of switch B0. This information is displayed in the status bar:

```
MCSSPR01.0.18 [31D] S FC SPAN[0/1] to SWCH B0.06
```
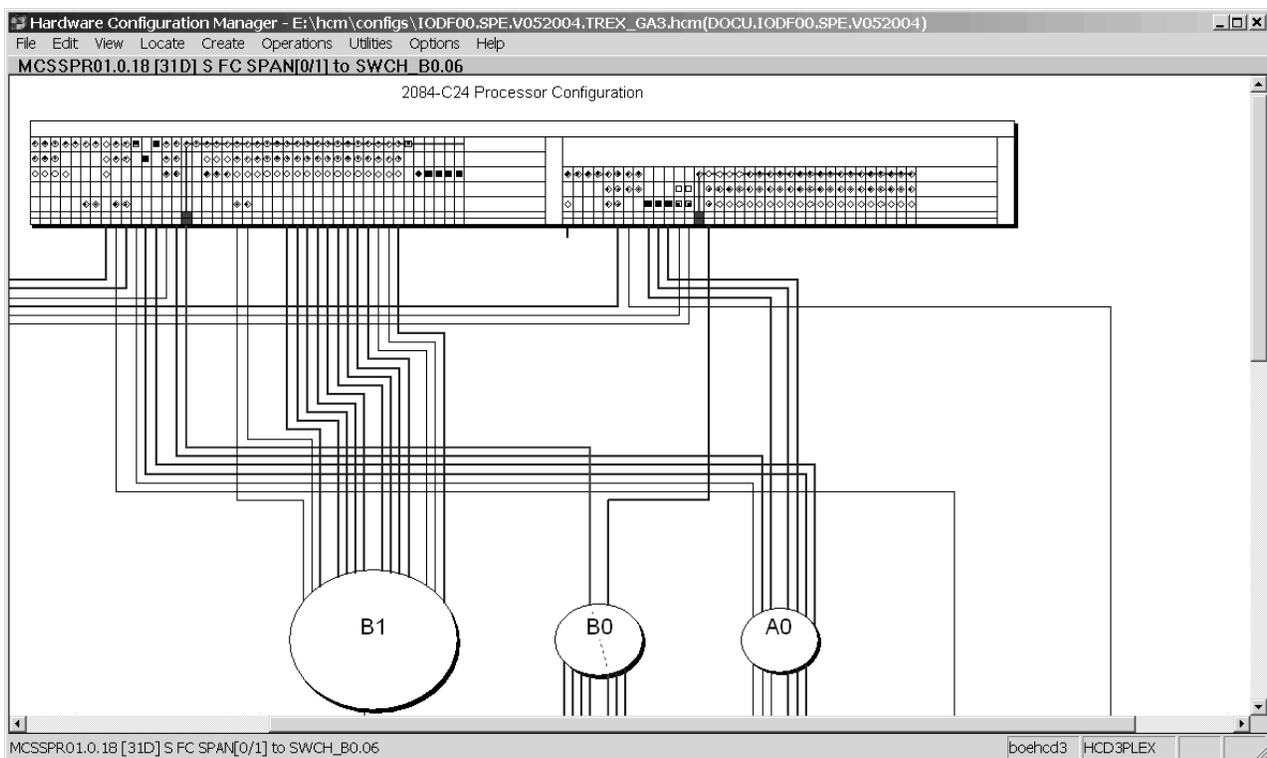

Figure 59. Spanned CHPIDs in the Configuration Diagram

In the configuration diagram, spanned CHPIDs are shown with one instance in each channel subsystem over which they are spanned. However, HCM ensures that

you can physically connect only the leftmost of these instances to a target. The configuration diagram shows which of the instances is physically connected. You can zoom into a processor or a channel subsystem in the diagram to see the following graphical hints as shown in the example from Figure 60:

- An arrow down (↓) indicates that this CHPID instance is physically connected to a target interface.
- An arrow to the left (←) or to the right (→) indicates that another CHPID instance of the current spanned CHPID is physically connected to a target.
- For dedicated, reconfigurable or shared CHPIDs, you see an empty field.
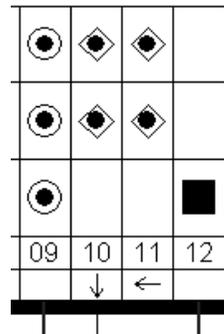
Figure 60. Physical connection information for spanned CHPIDs

Navigating with spanned CHPIDs

The pop-up menu for spanned CHPIDs in the configuration diagram contains selections with which you can navigate from the current instance to other instances of the spanned CHPID:

Locate Cable	navigates to the instance that is physically connected to a target object. This is always the leftmost instance.
Locate Left Instance	navigates to the next left instance.
Locate Right Instance	navigates to the next right instance.

Support of spanned CHPIDs in the HCM dialog

Besides the enhanced visualization of spanned CHPIDs in the configuration diagram, special handling is reflected and enhanced in various places in the dialog. You can find the related information where appropriate:

- "Making a standard connection" on page 166
- "Assigning CHPIDs to partitions" on page 110

Please also read the information contained in "Physical and logical aspects of spanned CHPIDs" on page 111.

Aggregating CHPIDs

If you need to combine attachments of several CHPIDs onto a single CHPID within the same processor, HCM offers you the **Aggregate CHPID** wizard under the **Utilities** menu. All CHPIDs must be connected to the same switch and use the same dynamic switch. See "Aggregating CHPIDs" on page 203 for a description of this wizard.

Controllers

You can create, edit or delete controllers by calling up the appropriate dialogs for the controller object in the configuration diagram. You can also physically connect and logically define controllers to processors; once the connection is made, you can add physical objects to it, such as patchports, crossbar switches, converters or switches.

There are different possibilities to open the **Controller** dialog:

- Double-click on a controller in the configuration diagram.
- Choose **Controller...** from the **Edit** menu, and select the desired controller from the list which appears. Note that you can rearrange and filter the list according to your needs.
- Press the **Controller...** button in the **Edit Switch** dialog, to edit the switch control units and devices. See "Editing switch port attributes" on page 138 for more details.

The **Controller** dialog lists all channel interfaces and their connections for the selected controller. You can access related dialogs to edit the controller or list the control units it contains.

You can also select a channel interface to connect to a target, edit the fiber link for a connection, or change the type of a channel interface.

Note that when you select a channel interface to connect to a target, pressing the **Connect...** button begins the process of connecting the controller and one or more processors. The **Connect Channel Interface** dialog which appears leads to other dialogs which allow you to create any necessary logical definitions for the physical connection. See "Connecting controllers to processors and switches" on page 163.

The **Controller** dialog also lists any strings/devices connected to the controller through its device interfaces. If I/O devices have no physical string (they are *integrated* into the controller), the "virtual strings" containing them are displayed (see "Editing a controller" on page 116). You can select a string to edit, or create a string under the selected controller. See "Strings" on page 124 for more information.
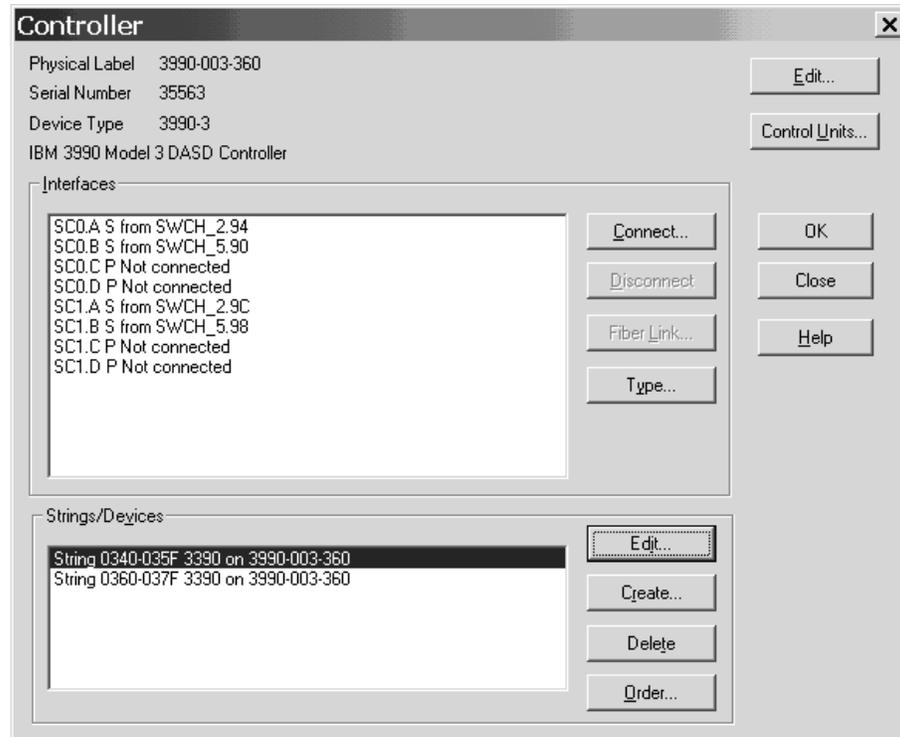
Figure 61. Controller Dialog

Creating a controller

To create a new controller, open the **Create Controller** dialog by choosing
Controller... from the **Create** menu. Enter the information for the controller and
press **OK**.

If there is more than one physical description of the control unit type, the **Choose
Physical Description** dialog will be opened.

If the physical description for the controller has more than one segment, the **Add
Control Unit** dialog will be opened for you to create control units for the other
segment(s).

The controller object will appear in the configuration diagram.

After creating the controller, you can edit it and connect its interfaces to objects,
using dialogs accessed through the **Controller** dialog. Additional control units may
be added to the controller through the **Create Control Unit** dialog.

Choosing a physical description

The physical description file, or *PDF*, is a template associated with the controller or
string that describes its physical characteristics. See Chapter 17, "Physical
description files (PDFs)," on page 373 for more details.

In the **Create Controller** dialog, if the control unit type selected has only one
possible physical description, then HCM simply creates the controller when you
press **OK**. If the control unit type selected has more than one possible physical
description, pressing **OK** causes the **Choose Physical Description** dialog to

automatically appear. (You can deliberately open this dialog; see "Editing a controller"). Select a physical description and press **OK** to create the new controller.

Changing the physical description affects all controllers within a subsystem (e.g. dual frame); you may only change to a physical description for a subsystem with the same or fewer number of components.

Adding control units to controller segments

If you are creating a controller with more than one segment in the **Create Controller** dialog, pressing **OK** will display the **Add Control Unit** dialog to allow you to create control units for the second (and subsequent) segment. You need only enter the control unit number and description; the serial number and type will be copied from the control unit in the first segment.

Ordering strings connected to the controller

In the **Controller** dialog press the **Order...** button in the Strings/Devices section. This opens an order dialog, allowing you to order the strings for a selected device interface. If there is more than one device interface on this controller, a dialog will be opened for you to choose on which device interface you want to re-order strings.

Editing a controller

Click on the **Edit...** button in the **Controller** dialog to open the **Edit Controller** dialog. Edit the information for the selected controller, or access related dialogs to add user data, change the physical description, arrange components within the controller, or group controllers into subsystems (for example, dual frame).

Note that if the Physically Installed check box is not enabled, the controller is un-installed, and drawn with dotted lines in the configuration diagram.

If the Devices are Integrated check box is enabled, all strings connected to the selected controller are integrated into the controller. Integrated strings and devices are not drawn in the configuration diagram, but can be accessed through the **Edit** menu or the Strings section of the **Controller** dialog. When you first create a controller, if its type is anything except DASD or Tape, it will be automatically marked as "devices are integrated".

The Display Class list box allows you to select a display class of the subsystem to which the controller belongs. The display class only affects the display by HCM of certain classes of controllers; the display class is initially set by HCM during IODF load/resync according to the first control unit added to the subsystem (see "Filtering your view" on page 48 for details).

Figure 62. Edit Controller Dialog

Adding user data to a controller

Add or edit user data by pressing the **Info...** button in the **Edit Controller** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Choosing a physical description

You can change the physical description of a selected controller by pressing the **Change Physical Desc...** button in the **Edit Controller** dialog. The **Choose Physical Description** dialog appears to allow you to select a different physical description for the controller. This dialog may automatically open during the creation of a controller; see "Creating a controller" on page 115.

Note that this dialog shows the PDF name for the subsystem to which the controller belongs, initially set during IODF load/resync according to the first control unit type. Changing the physical description affects all controllers within a subsystem (e.g. dual frame); you may only change to a physical description for a subsystem with the same or fewer number of components. If you want to change the PDF for only one controller in a subsystem, you must first ungroup the controllers (see "Grouping controllers into subsystems" on page 119).

Arranging controller components

During an IODF load or resync, HCM automatically places new control units inside controllers, creates channel and device interfaces, and connects them to the proper CHPIDs and strings. After the load or resync, you may need to correct the order of these components, or their assignment to controller segments.

Click on the **Arrange...** button in the **Edit Controller** dialog to open the **Arrange Controller** dialog.
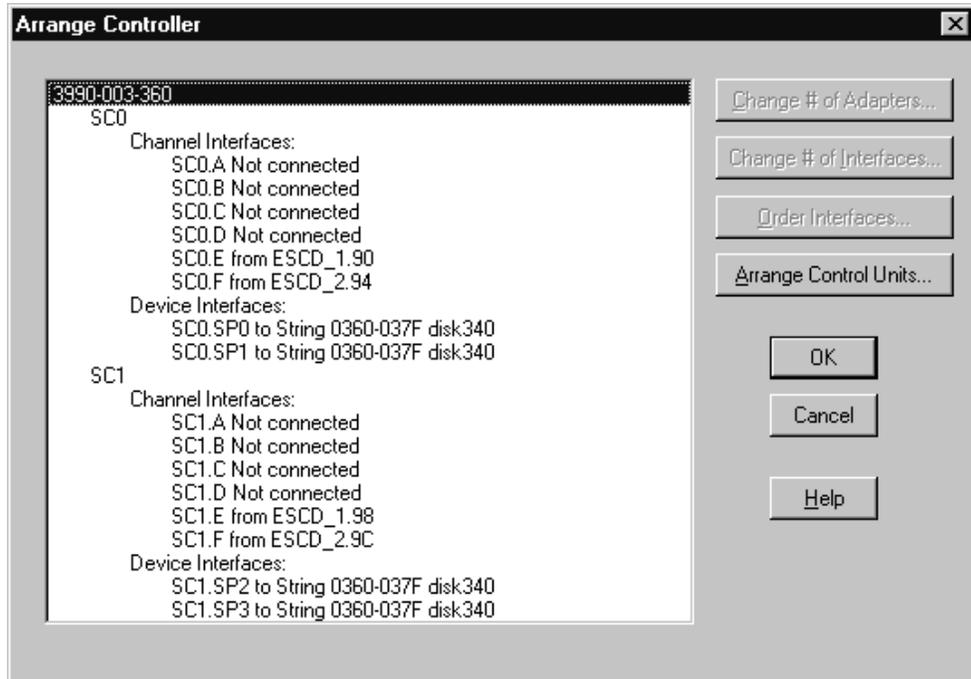
Figure 63. Arrange Controller Dialog

The list box shows all the physical components in the selected controller arranged hierarchically with sub-components listed below and indented from their containing components. See Figure 63 for an example.

Note that many types of controllers are partitioned into two separate sections; HCM uses the term segment to describe this architecture. For example, DASD controllers typically contain two segments.

Changing the number of interfaces

Select a segment (e.g. SC0) from the Arrange Controller list box and press the **Change # of Interfaces...** button. The **Number of Interfaces** dialog appears. You can select a number from the drop down box or type in the field. If you are adding interfaces, the new ones are added to the end of the segment and assigned labels from the physical description; interfaces over the maximum supported by the physical description will be assigned the label "INT". Note that new channel interfaces are created as serial interfaces if the first control unit in the segment is DASD or Tape, otherwise they are created as parallel interfaces.

Changing the order of controller interfaces

Select a segment from the Arrange Controller list box and press the **Order Interfaces...** button. The **Order Controller Interfaces** dialog appears, allowing you to move channel and devices interfaces from list boxes on the left (old order) to the list boxes on the right (new order). Use the >> and << buttons to move selected interfaces, or double-click on an interface to move it to the opposite list box. Note that you cannot leave this dialog without completely specifying the order for both channel and device interfaces; that is, either the left or right list boxes must be completely full.

Arranging control units and connections in a controller

Select a controller that contains more than one segment from the Edit Controller list box and press the **Arrange Control Units...** button. The **Arrange Control Units**

and Connections dialog appears to allow you to move the control units, the processor connections, and the string connections between the segments.
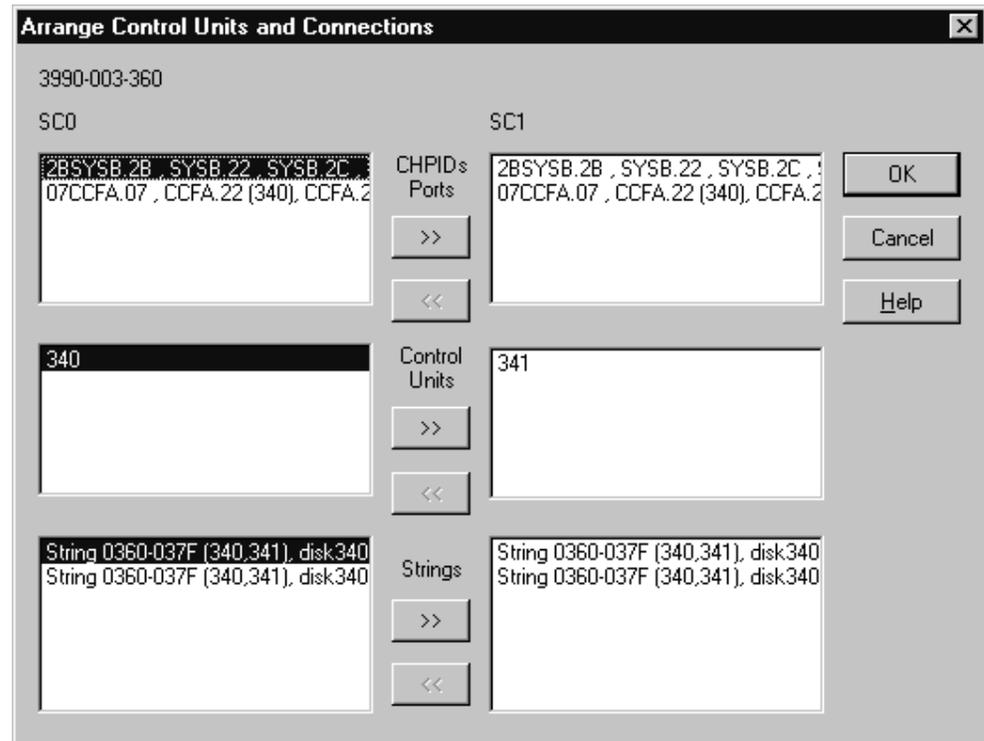
Figure 64. Arrange Control Units and Connections Dialog

You can move components between the segments by using the >> and << buttons to move selected items; alternatively, double-click on a component to move it to the opposite list box. After you have arranged the components, it may be necessary to use the **Order Controller Interfaces** dialog (see "Changing the order of controller interfaces" on page 118) to further correct the order of interfaces within each segment.

Grouping controllers into subsystems

Some controller types can be grouped together in a subsystem. For example, 3990 dual frame controllers or 3803s linked via communications cables

Click on the **Group...** button in the **Edit Controller** dialog to open the **Group Controllers** dialog (see Figure 65 on page 120). The Independent Controllers listbox lists all controllers of the appropriate type that are currently not grouped. You can move a selected controller into or out of a subsystem; HCM will make the appropriate physical changes. Use the >> and << buttons to move selected items; alternatively, double-click on a controller to move it to the opposite list box.
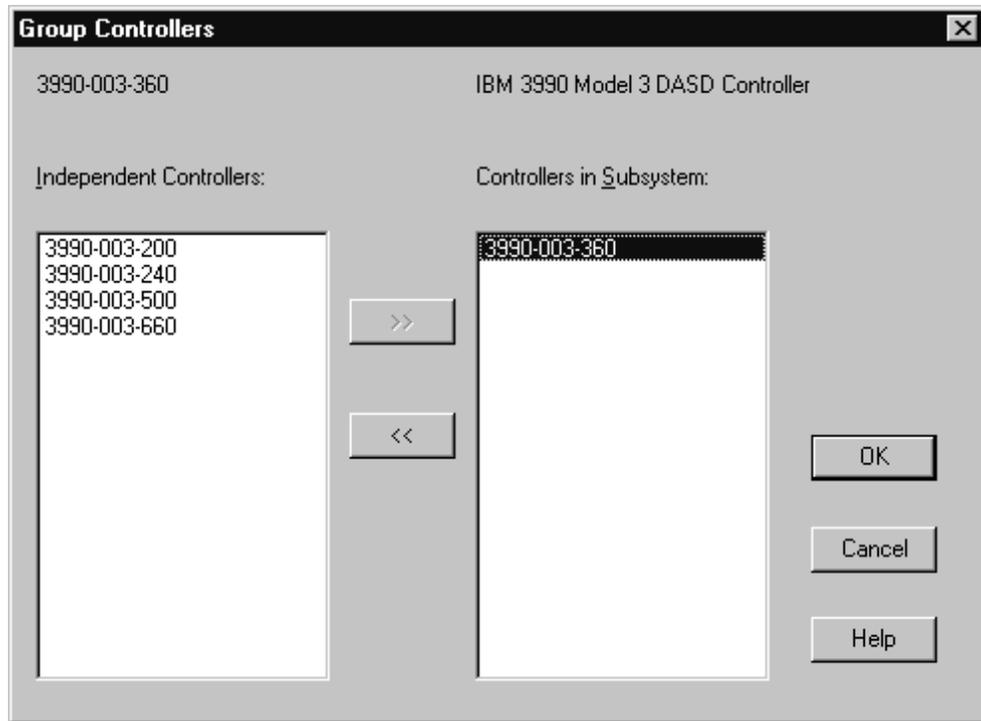
Figure 65. Group Controllers Dialog

Editing the fiber links above the controller

In the **Controller** dialog, select a channel interface and press the **Fiber Link...** button to open the **Edit Fiber Link** dialog. See "Editing fiber links in the dialog" on page 195 for information on connecting patchports.

Changing the channel interface type

Select a channel interface from the Interfaces list box in the **Controller** dialog and press the **Type...** button (see Figure 61 on page 115). The **Interface Attributes** dialog will appear, allowing you to make the selected channel interface either serial, parallel (bus and tag), IOC or OSA. See "CHPIDs" on page 103 for more details.
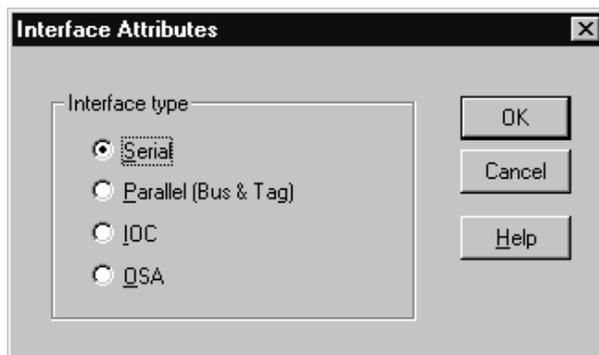
Figure 66. Interface Attribute Dialog

Deleting a controller

Select the controller object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the controller.

Control units

You can create or modify control units for a selected controller. There are two possibilities to open the **Control Unit List** dialog (Figure 67 on page 122):

- Select a controller by double-clicking on it in the configuration diagram, or by choosing the **Controller...** command from the **Edit** menu and selecting the desired controller from the list which appears. The **Controller** dialog will open for the selected controller; press the **Control Units...** button to display the **Control Unit List** dialog.

- Choose **Control Unit...** from the **Edit** menu and select the control unit name from the list that appears. Note that you can rearrange and filter the list according to your needs. Clicking **OK** causes the **Controller** dialog to open. In this dialog click the **Control Units...** button to display the **Control Unit List** dialog. This dialog lists the control unit you selected from the **Edit** menu.

In the **Control Unit List** dialog you can access related dialogs to create, edit or delete control units. You can view the devices connected to a control unit by selecting the control unit and pressing the **I/O Devices...** button to display the I/O Device List.

> **Note:**
> You can also access the **Control Unit <=> Processor Definition** dialog by pressing the **CU<=>Proc...** button. This dialog allows you to logically define the selected control unit to all or some of the processors that the controller can reach through physical connections.
>
> See "Making a standard connection" on page 166 for details.

You can modify the switch control unit by accessing the **Controller** dialog. For more details, see "Editing switch port attributes" on page 138.

The **Status** button allows you to retrieve information by control unit and to check system status.
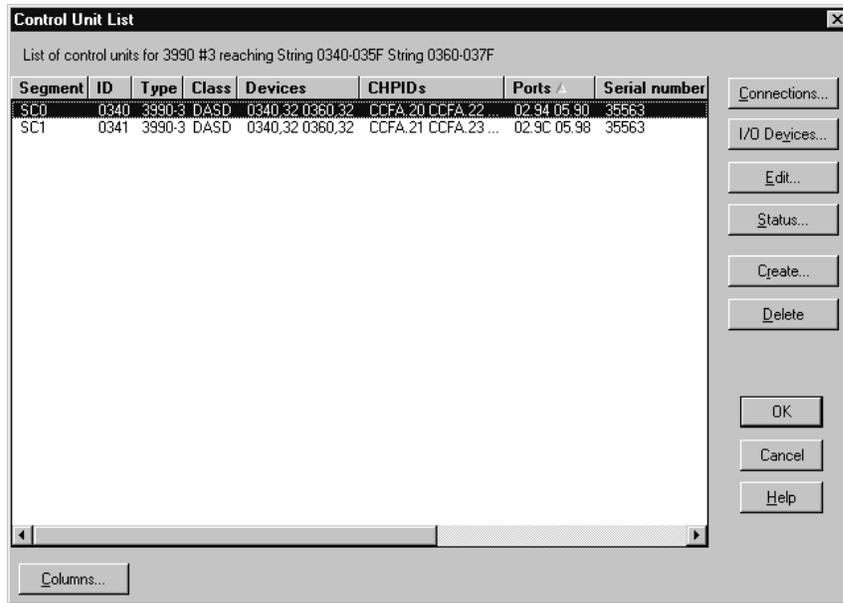
Figure 67. Control Unit List Dialog

Creating a control unit

You can create additional control units for a selected controller. Click on the **Create...** button in the **Control Unit List** dialog to display the **Create Control Unit** dialog. Enter the control unit information and press **OK**. Note that control units have the same serial number as the controller containing them; thus, the serial number shown in this dialog can be only edited in the **Edit Controller** dialog. The control unit is added to the controller in the diagram, and appears in the **Control Unit List** dialog for the controller.

You can also create one or more switch's control units inside a switch. When you create a new switch's control unit inside a switch, it is always connected to port FE of the switch. If you define an existing control unit for a switch, note that this control unit is added into the appropriate switch's *internal structures*. Internal structures of the switch are the switch's control units and devices. You must specify either both a control unit and a device or neither.

Choosing a controller segment to assign the control unit

If the controller has only one segment, any new control units created in the **Create Control Unit** dialog are placed in that segment. If the controller has more than one segment, pressing **OK** in the **Create Control Unit** dialog displays the **Choose Controller Segment** dialog. Choose the segment to which the new control unit will be assigned and press **OK**.

Note that you can assign the control unit to a different segment using the **Arrange Control Units and Connections** dialog (see "Arranging controller components" on page 117).

Editing a control unit

Select a control unit from the list box and click the **Edit...** button in the **Control Unit List** dialog. Alternatively, choose **Control Unit...** from the **Edit** menu and select the control unit name from the list that appears. Note that you can rearrange

and filter the list according to your needs. The **Edit Control Unit** dialog appears for the selected control unit. You can edit the control unit number, type, and description.

Edit the information and press **OK**. Note that you cannot edit the control unit serial number in this dialog; since it is the same as its controller, you can only edit it in the **Edit Controller** dialog.

Deleting a control unit

Select a control unit from the list box in the **Control Unit List** dialog and press the **Delete** button. HCM will display a confirmation dialog before deleting the control unit from the selected controller.

Deleting a switch's control unit

You can delete a switch's control unit from a switch. If the control unit has switch's devices defined, a warning message appears stating that HCM will cause implicit deletion of the devices. The reason for implicitly deleting the devices is that leaving a switch's device in an internal string creates a corrupt configuration. Devices are not allowed to reside in the internal string of a switch unless they are logically defined in the IODF to a switch's control unit which is defined to port FE of the switch.

Moving an existing switch's control unit into a switch

If your configuration contains control units which are meant to be internal to a switch but are not yet connected to the internal control unit port of the switch, you can move a control unit with its associated devices into a switch. To do this, click on the **Add Existing ...** button in the **Control Unit List** dialog.
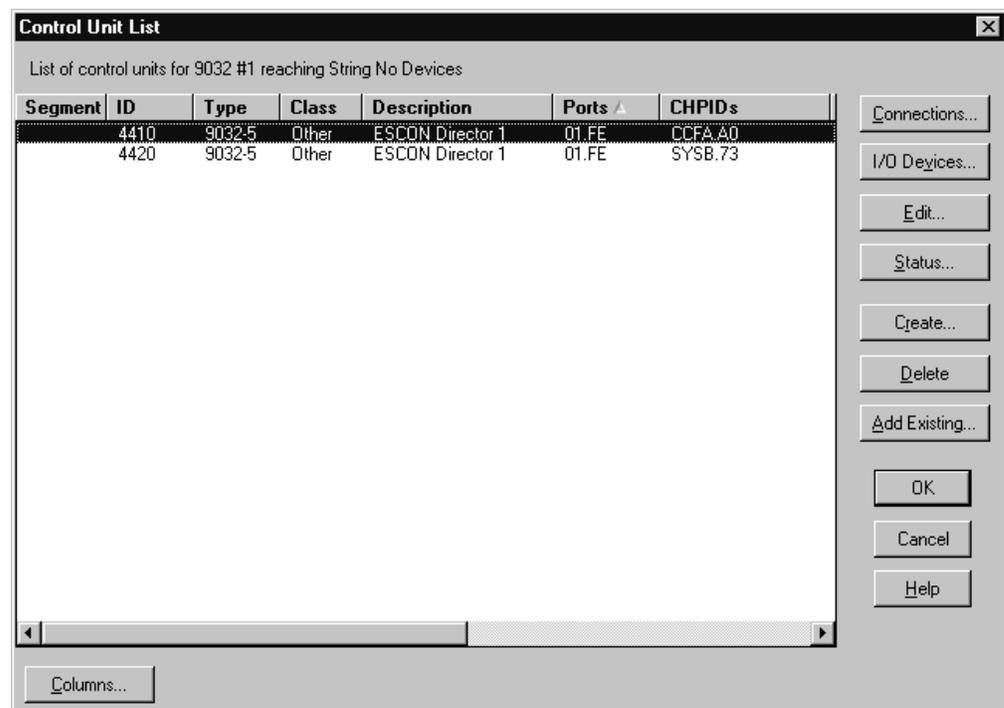
Figure 68. Control Unit List Dialog

The **Add Existing...** button is only displayed on the **Control Unit List** dialog when you reached this dialog via an internal controller. The **Add Existing Control Unit to Internal Controller** dialog is displayed.
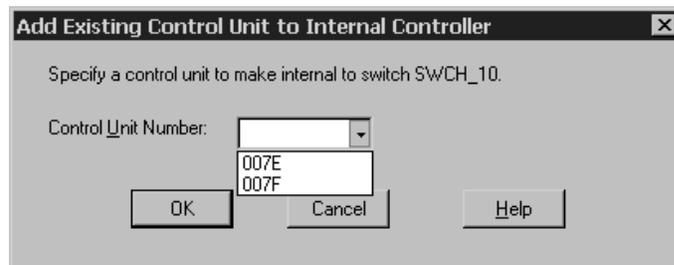
Figure 69. Add Existing Control Unit to Internal Controller Dialog

The drop-down box will display the numbers of all control units in the configuration with the same type as the switch which do not have any connections to switch ports. Thus, the drop-down box does not show control units which are already internal to switches. After your selection of the appropriate control unit, click on **OK** to connect the control unit to port FE of the current switch. You cannot move a control unit which is already connected to another switch port. If the selected control unit has devices under it, they will also be moved into the switch's internal string.

Strings

A *string* is a collection of one or more I/O devices. The term usually refers to a physical string of *units*, where a unit is a collection of I/O devices in a physical box; however, it may also refer to I/O devices integrated into a control unit. You can create, edit or delete strings, units, and devices by calling up the appropriate dialogs for the string object in the configuration diagram. You can also connect strings to controllers.

You can open the **String** dialog in different ways:
* Double-click on the top rectangle of a string in the configuration diagram.
* Choose **String...** from the **Edit** menu, and select the desired string from the list which appears. Note that you can rearrange and filter the list according to your needs.
* Double-click on a controller in the configuration diagram (or choose **Controller...** from the **Edit** menu) to display the **Controller** dialog; then select a string and press the **Edit...** button in the Strings/Devices section to open the **String** dialog.

The **String** dialog lists all string interfaces and their connections to controllers. You can access related dialogs to edit the string or its I/O devices, or connect a string interface to a controller device interface.

Selecting a string interface and pressing the **Connect...** button begins the process of connecting the I/O devices in the string to the controller, *and* any processors reachable through the controller. The **Select Controller Device Interface** dialog which appears leads to other dialogs allowing you to create any necessary logical definitions for the physical connection. See "Connecting devices to controllers" on page 157.
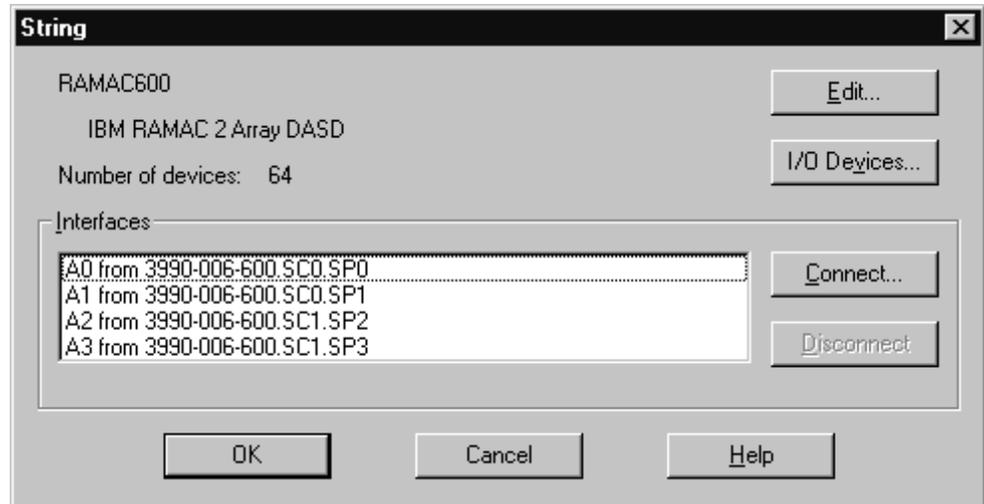
Figure 70. String Dialog

Creating a string

To create a string, choose **String...** from the **Create** menu to open the **Create String** dialog. Enter the string information and press **OK**. The string object will appear in the configuration diagram. You can then use the **String** dialog and related dialogs to edit the string, its units and its I/O devices, and to connect the string to a controller.
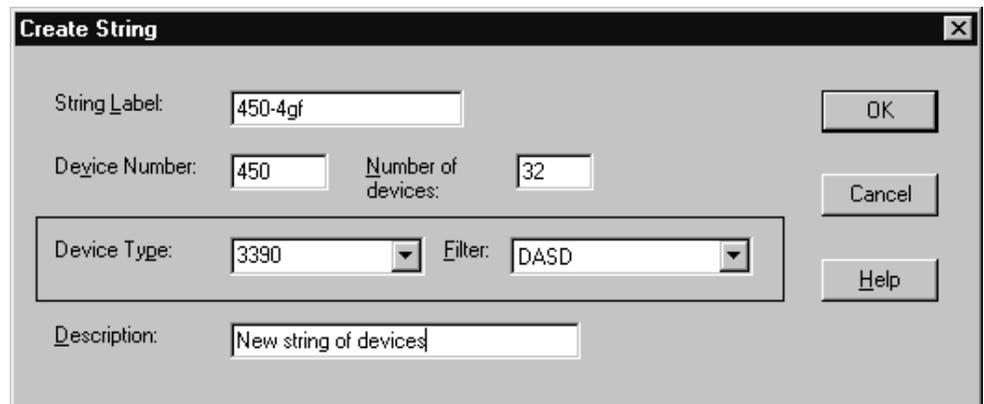
Figure 71. Create String Dialog

To create a string under a controller, first double-click on the controller in the configuration diagram (or choose **Controller...** from the **Edit** menu) to display the Controller dialog. Select a string and press the **Create...** button in the Strings/Devices section to open the **Create String** dialog. After you enter the string information in this dialog, pressing **OK** opens the **String** dialog to allow you to connect the new string to the selected controller (or another controller). You may choose not to connect the string, in which case the string is created, but it is not connected.

Note that serial numbers are assigned in the **Edit Unit** or **Edit I/O Devices** dialog; VOLSERs can be assigned in the latter dialog as well.

Choose physical description

The physical description file, or *PDF*, is a template associated with the controller or string that describes its physical characteristics. See "Graphical representation of physical objects" on page 71 for more details.

If there is more than one physical description possible for the string, pressing **OK** in the **Create String** dialog opens the **Choose Physical Description** dialog. Select a physical description and press **OK** to create the string. If the string was created under a controller, pressing **OK** will open the **Select Controller Device Interface** dialog to allow you to connect the string to the controller.
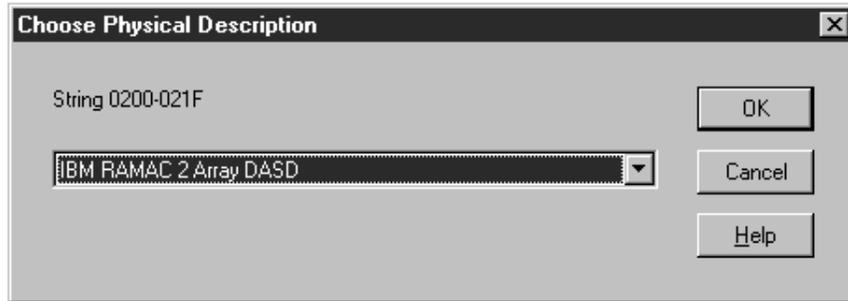
Figure 72. Physical Descriptions Dialog for the Unit Type 3390

Editing a string

Once you have created a string, you can edit it. Click the **Edit...** button in the **String** dialog to display the **Edit String** dialog. You can edit string information, add user data, or change the physical description for the string. You can assign devices to units within the string, or change the order or number of string interfaces. Note that you can also create, edit or delete units in the string; see "Units" on page 129 for more details.
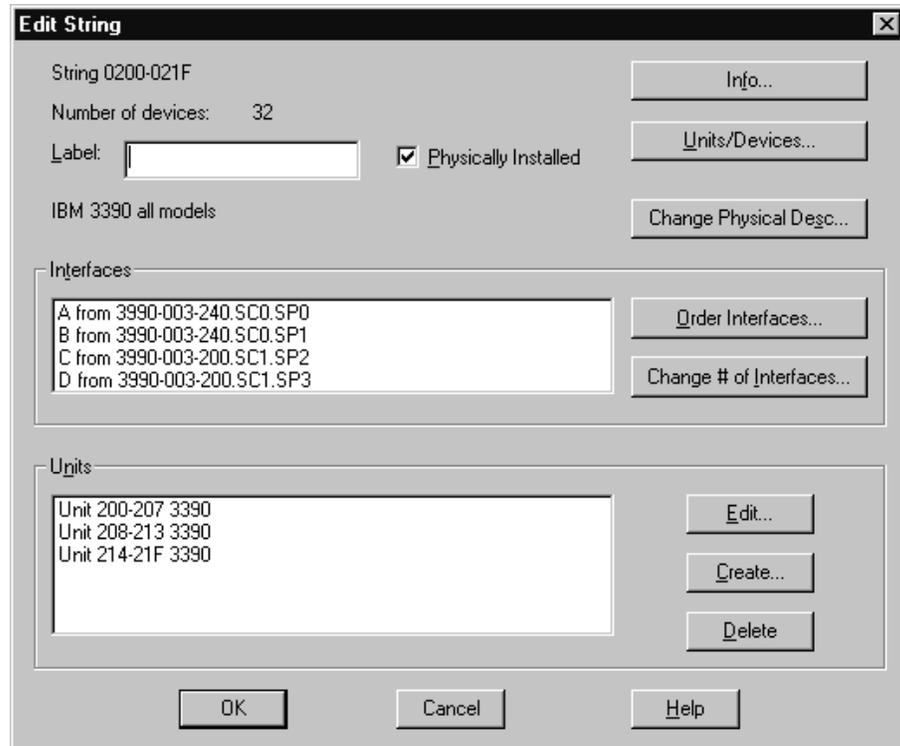
Figure 73. Edit String Dialog

Adding user data to a string

Add or edit user data by pressing the **Info...** button in the **Edit String** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Assigning devices to units in the string

Click on the **Units/Devices...** button in the **Edit String** dialog to open the **Units/Devices** dialog. Select one or more I/O devices from the listbox and press Assign... to open the **Assign to Unit** dialog.
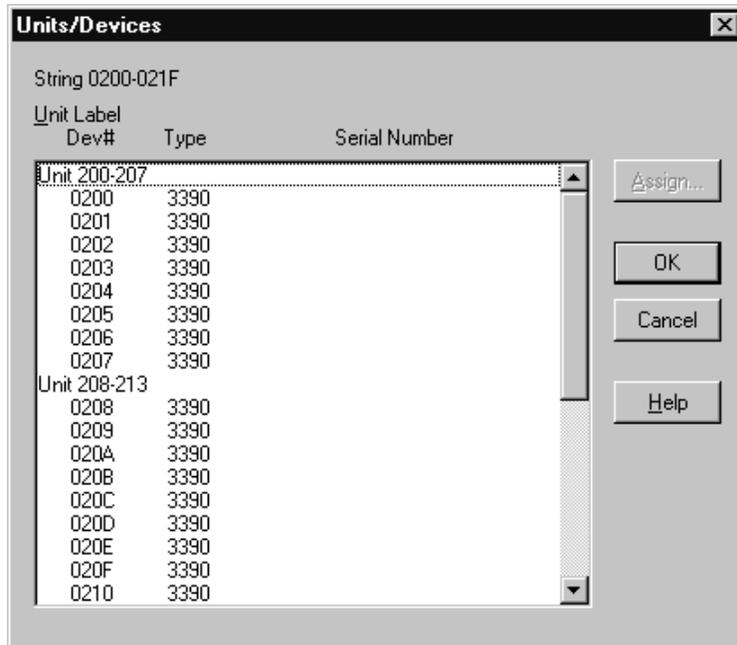
Figure 74. Units/Devices Dialog

Choose the unit to which the devices will be assigned. Use the checkboxes in the View box either to show only the units of this string or to show all units in all strings.

Changing the physical description

The physical description file, or *PDF*, is a template associated with the controller or string that describes its physical characteristics. See "Graphical representation of physical objects" on page 71 for more details.

You can change the physical description for a string. Click on the **Change Physical Desc...** button in the **Edit String** dialog to open the **Choose Physical Description** dialog. When you select a different physical description, the existing I/O devices under the string may be reassigned to units. New units are created as needed; there may be empty units left at the end of the string.

Ordering string interfaces

The string interfaces for a selected string are shown in the Interfaces section of the **Edit String** dialog. Click on the **Order Interfaces...** button to open the **Order String Interfaces** dialog. You can then re-order the string interfaces by moving them from the Current Order list box to the New Order list box. See "Ordering objects" on page 61 for an example of an ordering dialog.

Changing the number of string interfaces

Click on the **Change # of Interfaces...** button in the Interfaces section of the **Edit String** dialog. The **Number of String Interfaces** dialog appears to allow you to change the number of string interfaces.

Deleting a string

Select the string object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the string. Alternatively, you can select a string and its devices from the list and press **Delete** in the **Controller** dialog.

Units

A *unit* is a collection of I/O devices in a physical box; a string is a collection of units containing I/O devices. You can create, edit or delete units within a selected string. First, open the **String** dialog by double-clicking on a string in the configuration diagram, choosing the **String...** command from the **Edit** menu, or selecting a string and pressing the **Edit...** button in the Strings/Devices section of the **Controller** dialog. Then click on the **Edit...** button in the **String** dialog to open the **Edit String** dialog. This dialog lists all units in the selected string. Create, edit or delete units by selecting one or more units and pressing the appropriate buttons in the Units section of the dialog.

Creating a unit

Click on the **Create...** button in the Units section of the **Edit String** dialog. The **Create Unit** dialog appears; enter the information and press **OK** to create a new unit under the string. Note that you can add units before they are actually physically available; indicated whether the unit is physically installed by clicking in the check box.

You can assign I/O devices to the unit by clicking on the **Units/Devices...** button to open the **Units/Devices** dialog. See "Assigning devices to units in the string" on page 127.

Editing a unit

Select a unit and click on the **Edit...** button in the Units section of the **Edit String** dialog. The **Edit Unit** dialog appears, allowing you to edit the unit's serial number and label, and add user data. You can also indicate whether the unit is actually physically installed, and if the unit serial number should be applied to all devices within the unit. Note that if some of the devices have a different serial number, HCM will warn you that this information will be lost by applying the unit serial number to all its devices.
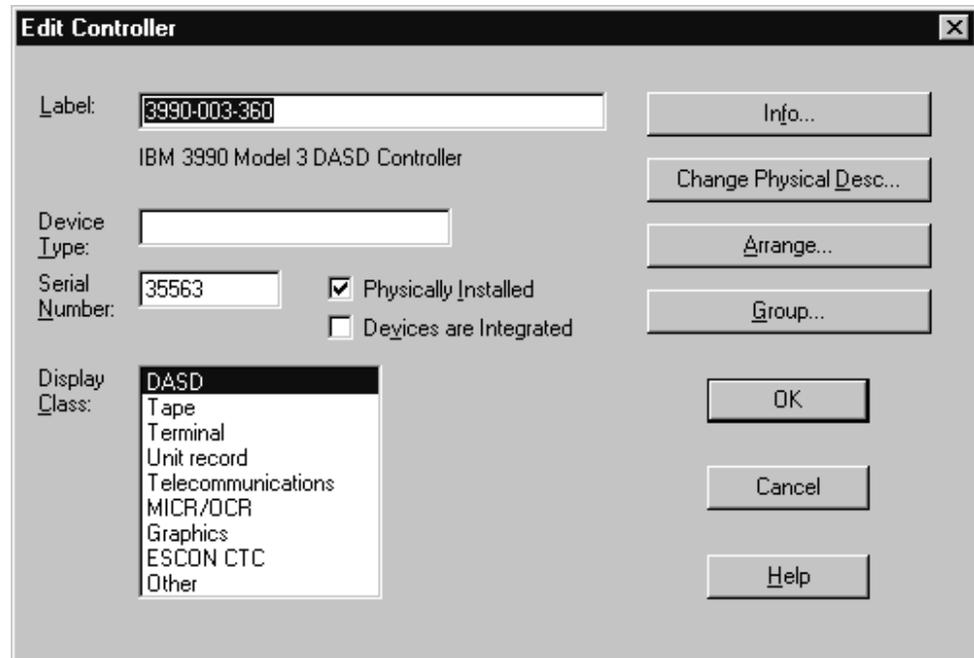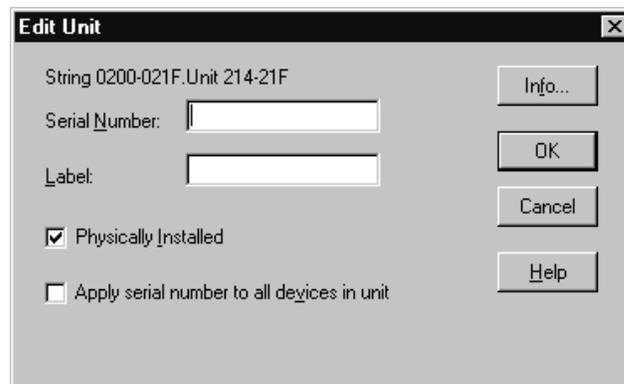

Figure 75. Edit Unit Dialog

Adding user data to a unit

Add or edit user data by pressing the **Info...** button in the **Edit Unit** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Deleting a unit

Select the unit in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the unit. Alternatively, you can select a unit from the list and press **Delete** in the **String** dialog.

I/O devices

I/O devices are grouped into units (physical boxes), which in turn are grouped into a string. You can create, edit or delete the I/O devices for a selected string. When the string is connected to a controller, the I/O devices can be logically defined through the physical connection not only to the controller, but also to any processors that are physically reachable through the connected controller. See "Connecting devices to controllers" on page 157 for details.

There are different possibilities to open the **I/O Device List**:
- Open the **String** dialog by double-clicking on a string in the configuration diagram, choosing **String...** from the **Edit** menu, or selecting a string and pressing the **Edit...** button in the Strings/Devices section of the **Controller** dialog. Then click on the **I/O Devices...** button in the **String** dialog.
- Press the **I/O Devices...** button in the **Control Unit List** dialog; note that the **I/O Device List** dialog will not display the **Create...** or **Delete...** buttons when called this way.
- Choose **Device...** from the **Edit** menu and select a device name or a range of devices from the list that appears. Note that you can rearrange and filter the list according to your needs. Clicking **OK** causes the **String** dialog to appear. In this dialog, click on the **I/O Devices...** button.
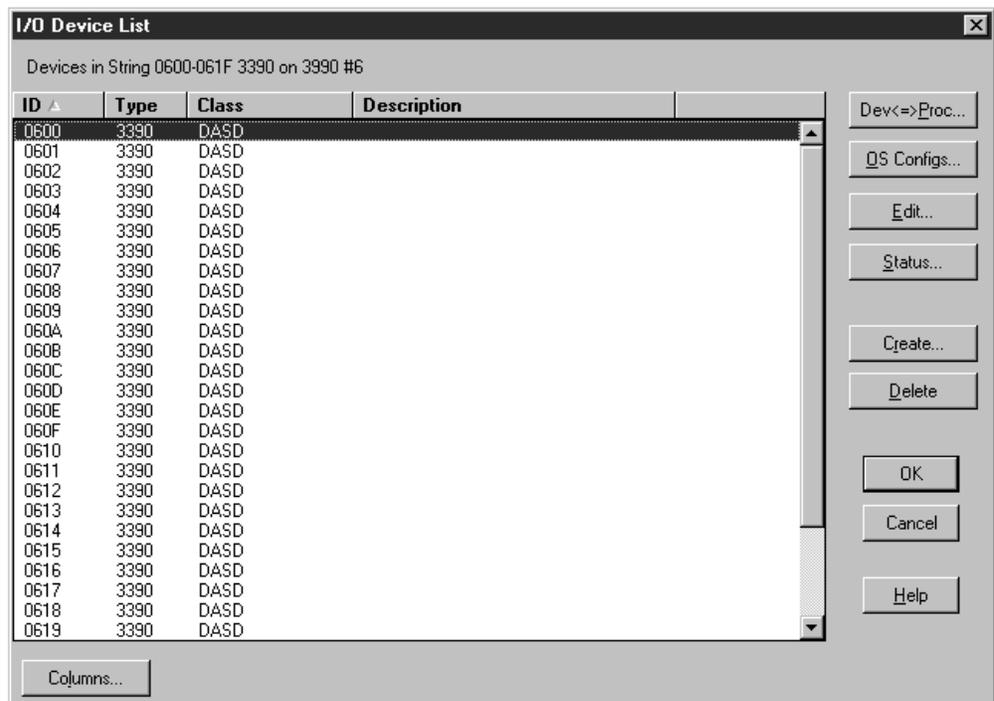
Figure 76. I/O Device List Dialog

The **I/O Device List** dialog allows you to display the serial numbers and VOLSERs, device descriptions, or control unit numbers in the list box by clicking the appropriate option button in the View section.

You can open related dialogs to create, edit or delete I/O devices by selecting one or more devices from the list box and pressing the appropriate button (as described below).

Note that you can access the **I/O Device <=> Processor Definition** dialog by pressing the **Dev<=>Proc...** button. This dialog allows you to select one or more processors from a list to which you define your selected device. For XMP processors, there will be one line per CSS in the list. See "Making a custom connection" on page 160.

Note: You can edit the device connected to the *switch control unit* by accessing the **Controller** dialog.

For more details, see "Editing switch attributes" on page 137.

Creating an I/O device in a string

Click on the **Create...** button in the **I/O Device List** dialog to open the **Create I/O Device(s)** dialog. Fill in the device information and press **OK**. Devices will be allocated to units appropriately. You can change the assignment of devices to units later. See "Assigning devices to units in the string" on page 127 for more information.
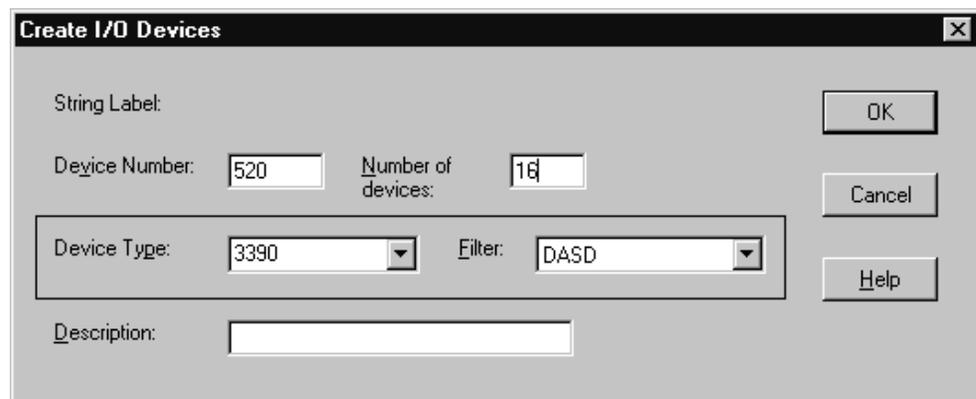
Figure 77. Create I/O Devices Dialog

When you click on **OK**, HCM creates the new devices and connects and defines them to all control units in the appropriate controller segment. This may create logical definitions between devices and connected processors, which you can later edit (see "Connecting devices to controllers" on page 157).

You can also create devices inside an internal string of a switch. Note that you can only define one switch's device at a time to any of the available switch's control units. When a device is created inside an internal string connect it immediately to a switch's control unit, otherwise the device would not be classified as internal and would be removed from the internal string on any future resynchronization with the IODF.

Editing an I/O device

Select one or more I/O devices in the **I/O Device List** dialog and click on the **Edit...** button. The **Edit I/O Device(s)** dialog appears; if you selected more than one device, any changes you make in this dialog will apply to all selected devices and the **Serial Number** and **Volume serial** fields will be disabled. Device serial numbers can be set for all devices in a unit by editing the unit.

Any changes to volume serials (VOLSERs) that you make with HCM cause an immediate update of the IODF. You can also edit the VOLSERs in stand-alone mode. See also "VOLSER reconciliation" on page 94.

Edit the information and click on **OK**.

Note: You cannot edit a device to change the device number. If it is necessary to change a device number, you must delete the device and create a new one with the desired number.

Defining devices to OS configurations

To define devices to OS configurations, select the device(s) and press the **OS Configs...** button. This opens the **List Device <=> OS Configuration Definitions** dialog. See "Listing OS configurations to which devices are defined" on page 212.

Deleting an I/O device

You can only delete an I/O device from the **I/O Device List** dialog, not from the configuration diagram. Select the device(s) in the list box and press the **Delete...** button in the dialog. HCM will display a confirmation dialog before deleting the device(s).

Converters

There are two types of *converters*: those that are used to connect serial ESCON cables to parallel bus-and-tag cables (9034 or S->P), and those that connect parallel to serial cables (9035, or P->S). HCM automatically creates and connects *generic converters* during an IODF load or resync and when you connect a serial to a parallel interface (or vice versa).

Generic converters can be replaced with "real" converters that can be assigned a label, serial number, user data, and graphical position on the diagram. HCM maintains an inventory of "real" converters that can replace generic converters in your configuration.

Choose **Converter...** from the **Edit** menu to display the **Converters** dialog, that list converters and their attributes. Note that you can rearrange and filter the list according to your needs. Real converters that are in use can be edited but not deleted. Unused converters can be deleted.
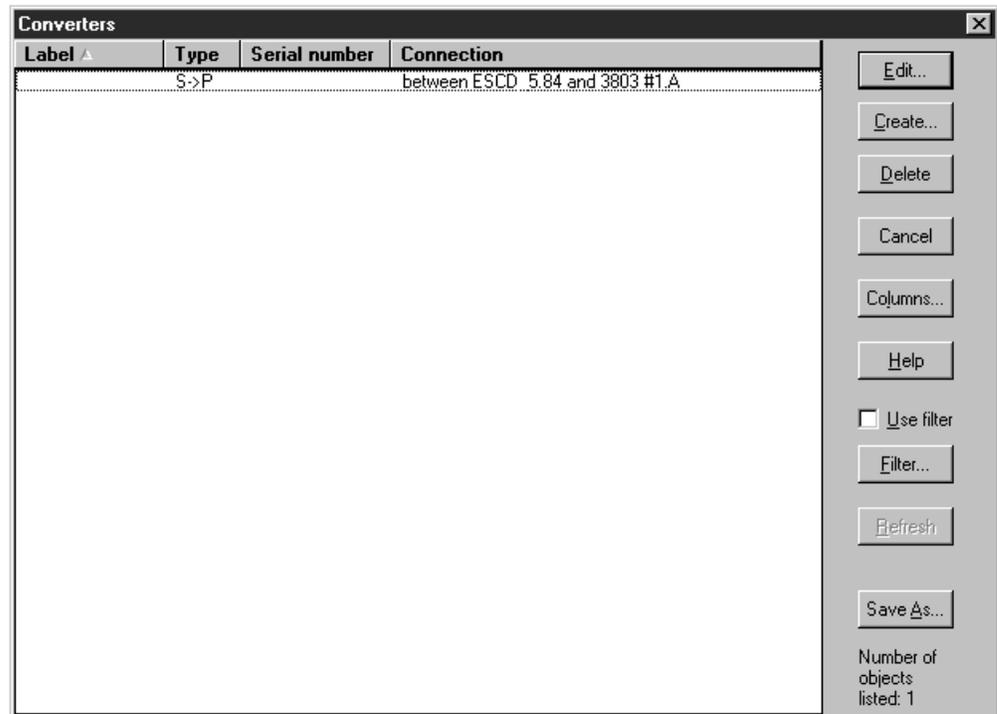
Figure 78. Converters Dialog

Creating a converter

In the **Converters** dialog, press **Create** to open the **Create Converter** dialog.
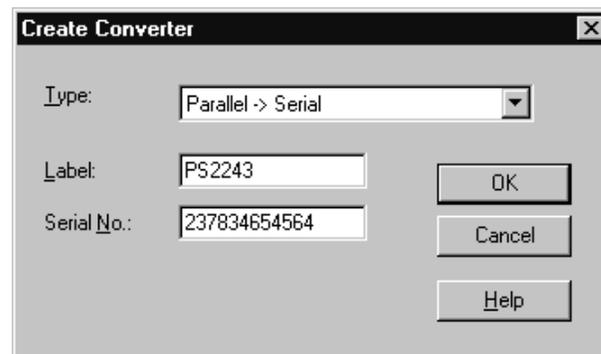
Figure 79. Create Converter Dialog

Select the converter type (P->S or S->P) and enter the label and serial number of
the converter. At least one of label or serial number must be specified. Press **OK** to
create the converter and return to the **Converters** dialog. The new converter can
now be used to replace a generic (or real) converter in your configuration.

Deleting a converter

In the **Converters** dialog, select the View option Unused Converters then select the
converter(s) that you want to delete in the listbox and press **Delete**. You cannot
delete converters that are in use.

Editing a converter

Open the **Edit Converter** dialog by double-clicking on a converter in the configuration diagram, or by pressing the **Edit...** button in the **Converters** dialog.

You can edit the fiber link above this converter, if it is an S->P converter, by pressing the **Fiber Link...** button. See "Editing fiber links" on page 192.

You can replace this converter with an unused converter or a newly created one by pressing the **Replace...** button. See "Replacing a converter" below.

You can edit the label, serial no., user data, and graphical position of this converter by pressing the **Edit...** button. This opens the **Edit Converter** dialog.
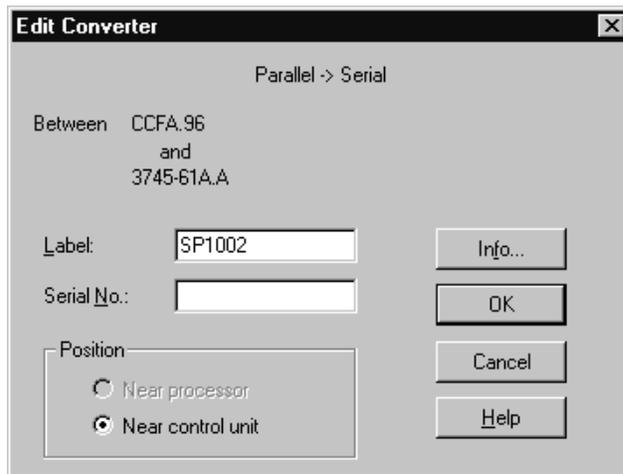
Figure 80. Edit Converter Dialog

Note: If this is a generic converter, then you can only edit its position.

Replacing a converter

To replace a converter with an unused converter or a newly created one, press the **Replace...** button in the **Converter** dialog. This opens the **Replace Converter** dialog.
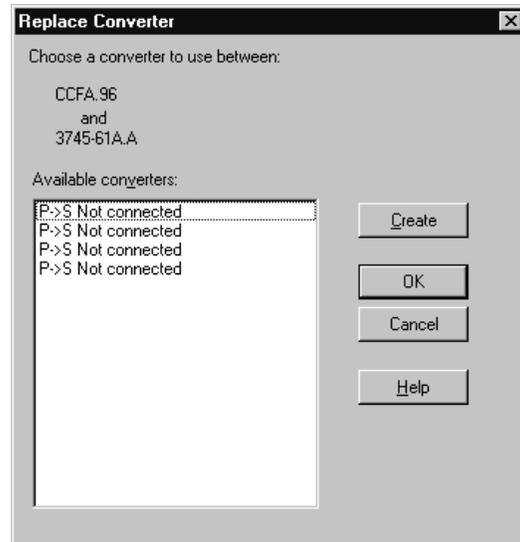
Figure 81. Replace Converter Dialog

The Available converters listbox lists all the unused converters of the appropriate type. To replace the converter with one of these, select the replacement and click **OK**. To create a new replacement converter, press **Create**. This opens the **Create Converter** dialog (see "Creating a converter" on page 133).

Switches

For switches you can dynamically restrict internal connections by specifying a switch configuration. The *switch configuration* is a set of connection rules used to determine which switch ports are connected at any one time. That is, each switch configuration deems that only certain intersections between switch ports are active. Any switch port may connect to any other switch port. HCM allows you to create, view, edit or delete switch configurations. You can also visualize switch configurations by defining a certain switch configuration for each switch. See "Visualizing ESCON or FICON switch and crossbar switch configurations" on page 51 for information on how to visualize switch configuration.

A switch can be connected through its switch ports to objects above it (processors), beside it (other switches), or below it (controllers or crossbar switches). All switch connections are either ESCON or FICON. See Chapter 7, "Connecting objects in your configuration," on page 155 for more information.

The switch contains *switch control units* which can receive and interpret processor messages to set the switch configuration dynamically. They are always connected to the switch port with the address FE. Each switch control unit is connected to a device, also within the switch. The switch control units and devices do not appear as objects in the configuration diagram. But the reports that are produced via the **Print Reports...** menu, report more than one switch's control unit and device per switch as appropriate. And access is provided for editing and viewing purposes via the **Edit...** button on the **Switch** dialog.

It is possible to specify a switch without, with one and with multiple switch's control units and devices. The switch's control units and devices are also called internal controller, internal string or internal structures.

You can create, edit or delete switches and their ports, control units, devices and fiber links. You can also manage switch configurations for a selected switch.

Note: HCM supports a generic FC switch (type FCS) supporting port addresses 00 to FF. This switch type does not support a switch control unit and switch device and therefore cannot be accessed by I/O Operations functions like migrate or activate switch configuration.

Creating a switch

Choose **Switch...** from the **Create** menu to open the **Create Switch** dialog. Fill in the information and press **OK** to create a new switch. Note that switch ports will be automatically created and assigned initial addresses based on the type and model of the switch specified.

Note: The default for installed switch ports is the minimum range, as specified in the appropriate UIM. To install additional ports, see "Editing switch attributes" on page 137.

When you create a switch, the following criteria for the switch's control units and devices must be met:
- If either a control unit number or a device number is specified, both must be specified.
- If existing control unit numbers and device numbers are specified, the control units and devices must already be connected to one another.

After you have created a switch, you can edit its attributes, ports, control units and devices. You can also connect its switch ports to processor CHPIDs or other switch ports. To connect the switch to controllers below, you must go through the controller dialogs.

Editing a switch

Open the **Switch** dialog by choosing **Switch...** from the **Edit** menu and selecting a switch from the resulting list. Note that you can rearrange and filter the list according to your needs. Alternatively, double-click on the desired switch in the configuration diagram. You can view a selected switch's installed ports and connections in this dialog. You can access related dialogs to edit switch or port attributes, or connect the ports to target interfaces such as CHPIDs or other switch ports. You can also open dialogs to edit the switch configurations for the switch, which determine its internal connectivity at any time.

The plus sign beside the control unit number and the device number in Figure 82 on page 137 is displayed if there are more than one switch's control units or devices defined to a switch. The switch's control unit number shown is always the one which is numerically the lowest. The switch's device number shown is always the one which is defined to the switch's control unit shown. If the displayed switch's control unit has no switch's device defined to it, no device number but only the plus sign is shown to indicate that there are switch's devices defined for the switch, but not for the shown switch's control unit.

Notes:
1. By pressing the **Port Status...** button you get the status of a port, whereas pressing the **Status** button provides you with the switch status.
2. Pressing the **Save Port List...** button lets you save the information from the **Ports** list into a text file.
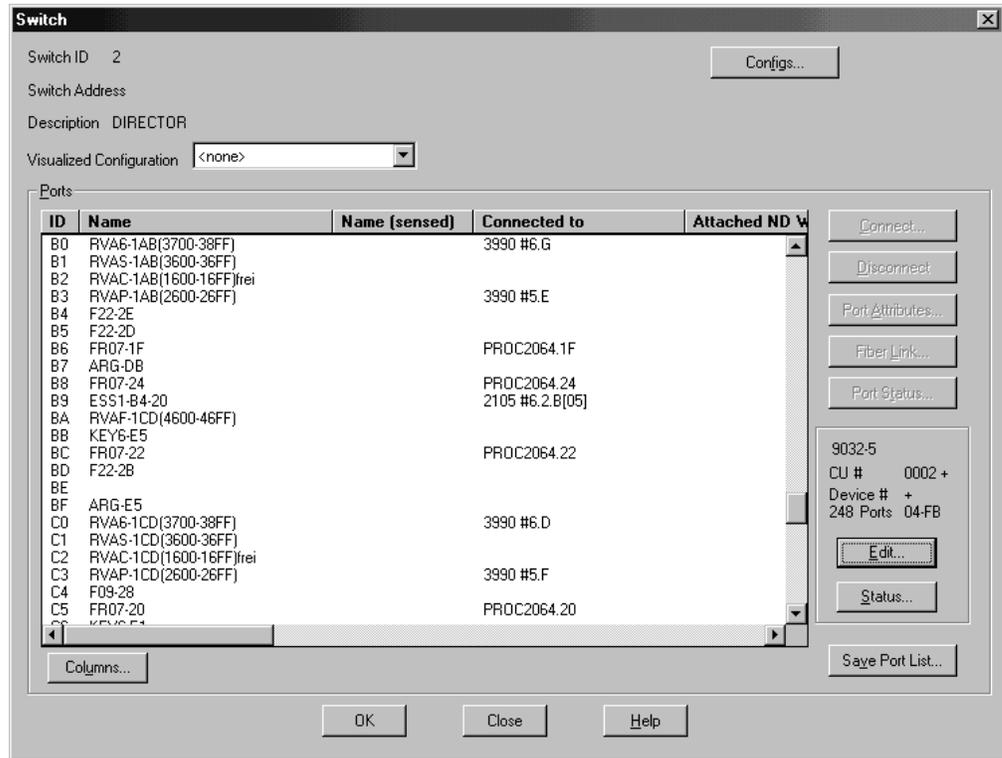
Figure 82. Switch Dialog

Editing switch attributes

Click on the **Edit...** button in the **Switch** dialog to open the **Edit Switch** dialog, which allows you to edit switch attributes as well as to view the switch's port, control unit and device characteristics.
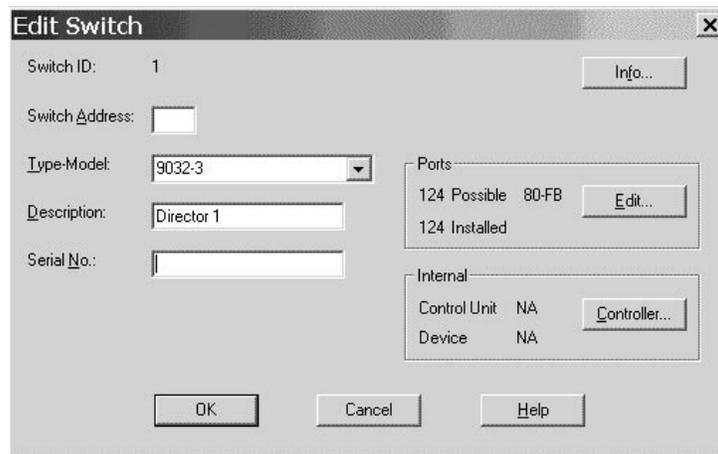
Figure 83. Edit Switch Dialog

From this dialog, you can access related dialogs to add user data, install or remove switch ports, edit the switch's control units or I/O devices, and set up connections to the control units.

Add or edit user data by pressing the **Info...** button to display the standard user data dialog. For more details, see "User data fields" on page 44

Click on the **Edit...** button in the Ports section to open the **Switch Ports** dialog. You can then select one or more ports from the list box to install or remove (uninstall). HCM assigns all installed ports the type indicated in the Newly Installed Ports Are... section: either Multimode (LED) or Single-mode (Laser).

The switch's control units and attached devices are displayed in the Internal section. Clicking on the **Controller...** button opens the **Controller** dialog. From here, you can access related dialogs to create or edit the switch control units and devices. See "Controllers" on page 114 and "I/O devices" on page 130 for more details.

As in the **Switch** dialog in Figure 82 on page 137 the plus sign in the Internal section indicates that there are more than one switch's control units or devices defined to the selected switch.

The **Status** button displays the switch status, and the **Port Status** button displays the port status.

Editing switch port attributes

Select one or more switch ports from the list box and click on the **Port Attributes...** button in the **Switch** dialog. The **Port Attributes** dialog appears, allowing you to edit the attributes of the selected switch port(s). If you select more than one port, this dialog appears for you to edit each in succession.
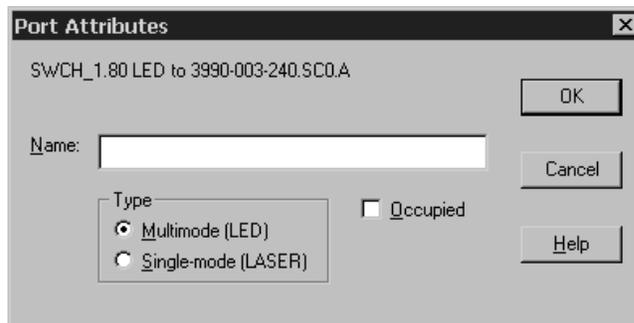
Figure 84. Port Attributes Dialog

Editing fiber links above the switch

Select one or more switch ports connected to a target serial interface such as a CHPID and click on the **Fiber Link...** button in the **Switch** dialog. The **Edit Fiber Link** dialog will appear, allowing you to edit the composition of the fiber link. See "Editing fiber links" on page 192 for more information.

Moving Ports

If you need to move CHPID-to-control-unit connections from one switch to another or need to rearrange such connections on a single switch, HCM offers you the **Move Port Connections** wizard under the **Utilities** menu. Refer to "Moving port connections" on page 201 for a description of this wizard.

Deleting a switch

Select the switch object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the switch. The switch's control units and devices are deleted as well.

See also "Deleting a switch's control unit" on page 123.

Managing switch configurations

Open the **Configurations for Switch XX** dialog by clicking on the **Configs...** button in the **Switch** dialog. You can then select a switch configuration associated with the specified switch to edit or delete, you can create a new configuration, or you can edit a port matrix.

For ESCON and FICON switches, the configurations can be stored either in the IODF or on the switch itself. For z/OS, you can work with the configurations stored on the switch, provided the ESCON Manager or I/O Operations is installed and running (see Chapter 11, "User interface to I/O Operations," on page 221).
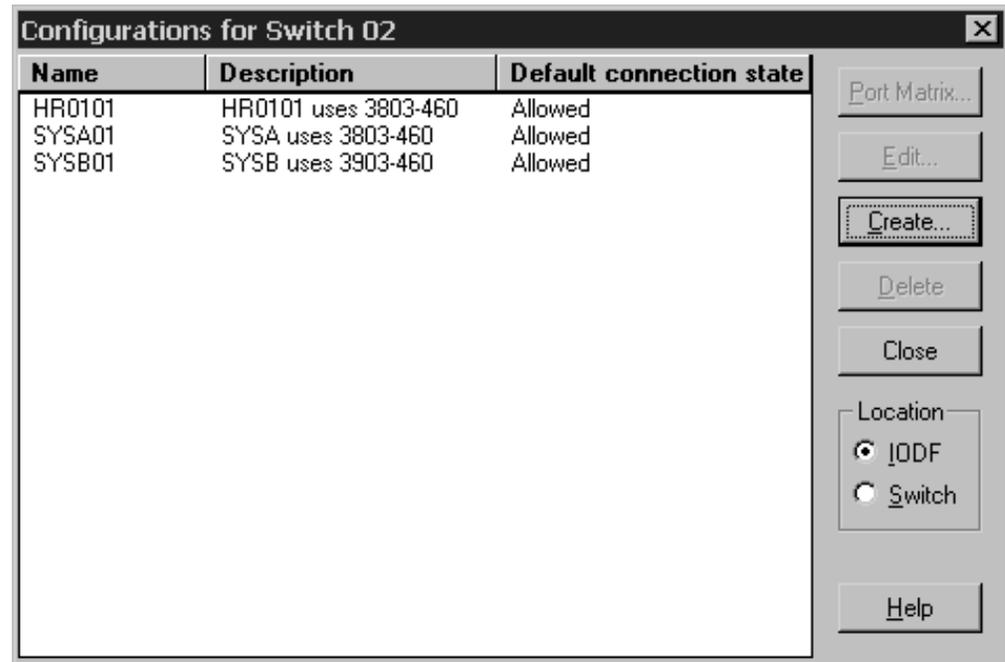
Figure 85. Configuration for Switch XX Dialog

Creating a switch configuration

Click on the **Create...** button in the **Configurations for Switch XX** dialog to open the **Create Switch Configuration** dialog. You can set the default connection state to be *allowed* or *prohibited*.

If the radio button **IODF** is active in the **Location** field, the **Create Switch Configuration** dialog will include a checkbox **Generate default matrix**. This check box is disabled if **Switch** is active instead.

When generating switch port matrices, all defined logical paths will be analyzed by HCD and in the case of chained connections, all possible paths of chaining switches will be determined by the visualized configurations.

Editing a switch configuration

Click on the **Edit...** button in the **Configurations for Switch** *XX* dialog to open the **Edit Configuration for Switch** *XX* dialog. You can change the default connection state to be *allowed* or *prohibited*.

Editing a port matrix

Open the **Edit Port Matrix** *XX* **for Switch** *YY* dialog by selecting a switch configuration from the list box and clicking on the **Port Matrix...** button in the **Configuration for Switch** *XX* dialog. You can also double-click on a switch configuration.
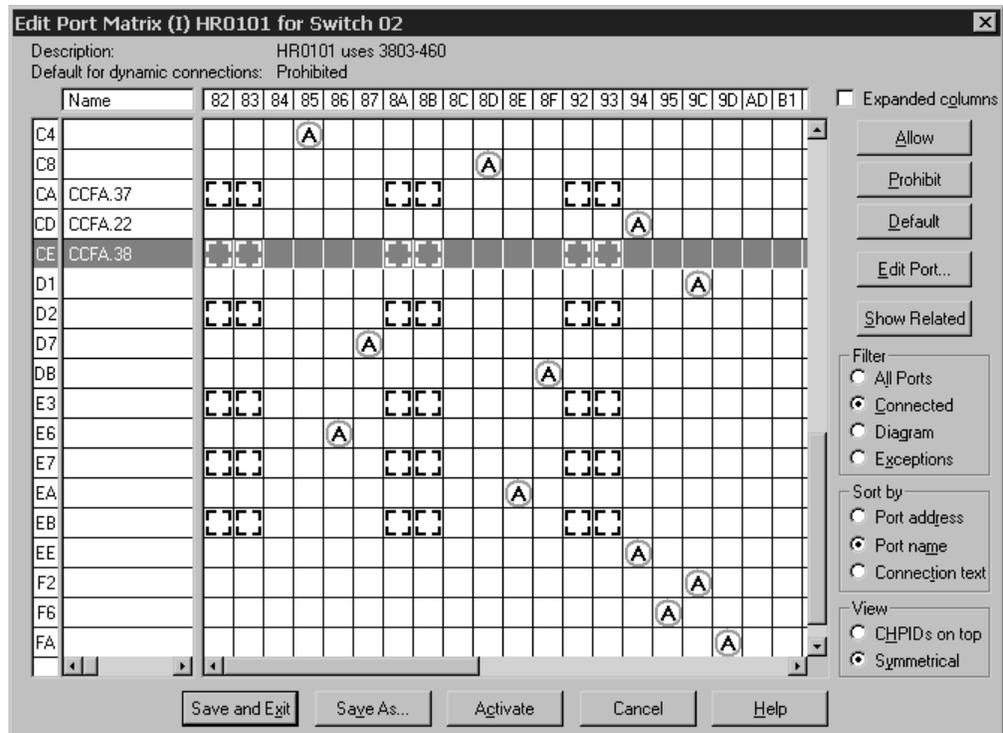
Figure 86. Configuration Port Matrix

The large scrollable window to the right is the *configuration port matrix*. Each row or column in the matrix corresponds to a port; each cell represents the intersection between a row port and a column port.

The scrollable window to the left of the port matrix is the port attributes window, displaying the attributes for each switch port (each row in the matrix). The columns in this window display the port name, whether the port is blocked or dedicated, and the identity of the CHPID or control unit (if any) connected to the port. This window will scroll vertically with the port matrix window.

You can drag the gray separator between the two windows to alter their relative widths.

You can select individual cells or you can select ports by clicking on the row or column header. Select a number of cells or ports by holding the Shift or CTRL key while clicking, or by clicking and dragging. Besides the window functions, you can also use the keyboard (arrow keys, page-up, page-down...) to move around the port matrix. In order to get more information about a port intersection or port

heading, keep the cursor over an intersection or a heading. After a short delay a small pop-up window appears that shows more detailed information (for example, how a port is connected).

Cells may contain different symbols with the following meaning:

Table 10. Symbols in Port Matrix

Symbol	Description
Black square	These ports have a dedicated link to each other.
Hollow square, hollow circle	There is a logical connection defined between the control unit and the CHPID connected to the ports of this intersection. Square: A connection requires a dedicated link. Circle: The intersection is used for a dynamic connection. A green color indicates that the port matrix data allows a connection, a black color indicates that the port matrix data does not allow a connection.
Broken square, broken circle	These symbols are used if there are several possible paths for control unit to CHPID connections through chained switches. Square: A connection requires a dedicated link. Circle: The intersection is used for a dynamic connection.
Letter A or P, or blank	These symbols indicate explicit "allows" or "prohibits" of dynamic connections. A: Dynamic connections are allowed. P: Dynamic connections are prohibited. Blank: The default connection is used.
Speckled background	Either the row port or the column port is blocked. If a port is blocked, the whole row or column appears with speckled background.

Click on the **Allow** or **Prohibit** buttons to change the connection state of all selected cells in the port matrix to *allowed* or *prohibited*, respectively.

Clicking the **Default** button changes the connection state of all selected cells in the port matrix window to whatever the default is set. To set the default connection state, click on the **Edit** button in the **Configurations for Switch *XX*** dialog.

Changing the view of the matrix: HCM allows you to filter and rearrange the view of the matrix according to your needs.

Click on the **Expanded columns** check box to enlarge the column header and show abbreviated text indicating to what the port is connected.

The radio buttons of the **Filter** group allow you to reduce the number of ports that will be shown in the matrix. Working in conjunction with the Filter radio buttons is the **Show Related** push button that operates on selected ports. To enable this button select one or more ports by clicking on the row or column heading. When you click the **Show Related** push button, HCM filters the matrix to show only those ports that have intersections of interest with the selected ports. An intersection of interest is one that contains a logical control unit to CHPID

definition or a dedicated connection. Clicking the now available **Restore Filter** button returns the matrix to the view when the **Show Related** button was clicked.

The radio buttons of the **Sort by** group controls the order of the ports in the rows and columns.

Use the radio buttons under **View** to rearrange rows and columns. The **Symmetrical** radio button shows the matrix in the traditional symmetrical format. Thus, each port is shown twice, once in row and once in column. The **CHPIDs on top** radio button shows ports connected to CHPIDs as columns and ports connected to control units as rows.

The following figure shows a configuration matrix that is filtered to show only those ports that are connected and that has the **CHPIDs on top** radio button selected. The columns are expanded and the cursor rested over an intersection to show more information.
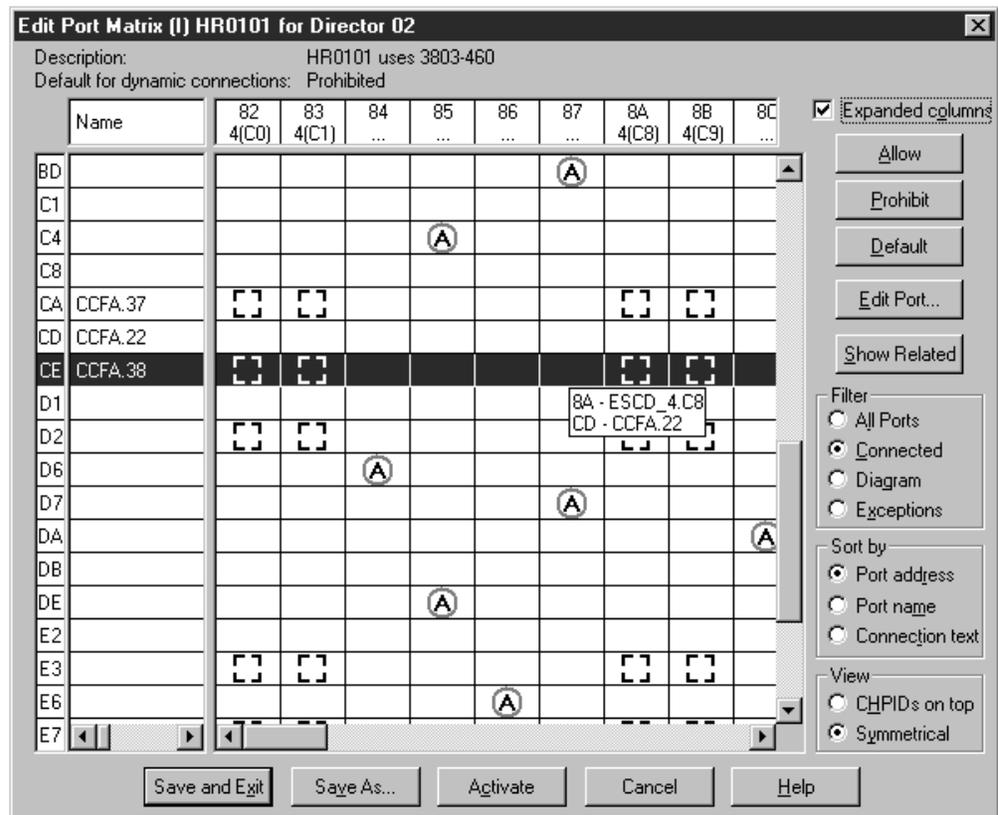
Figure 87. Configuration Port Matrix - Filtered and CHPIDs on Top

Visualizing a switch configuration

HCM can evaluate information from switch configurations. The diagram can show dedicated connections or blocked ports. In addition, when highlighting objects in a diagram, HCM can use the switch configuration data to show what particular route is used for CHPID-to-control-unit definitions (see "Visualizing ESCON or FICON switch and crossbar switch configurations" on page 51 for more detailed information).

To visualize a switch configuration, you first have to associate one particular switch configuration to each switch. Select **Visualized Matrices...** from the **View**

menu. In the following **Define Matrix Set** dialog, you can associate one switch configuration to each switch. Alternatively, you can select a configuration for a specific switch in the **Switch** dialog. You can save these definitions and display them later on again using **Save Named Matrix Set** and **Restore Named Matrix Set** under the **View** menu.

Editing switch ports in a configuration

Open the **Edit Switch Port** dialog by selecting one or more ports and clicking on the **Edit Port...** button in the **Edit Port Matrix** *XX* **for Switch** *YY* dialog. If more than one port is selected, this dialog will appear for each port in succession. Alternatively, you can double-click on a port in the port attributes window of the **Edit Port Matrix** *XX* **for Switch** *YY* dialog.

You can block the selected switch port, such that it can never be connected while blocked. Blocked ports are indicated as rows of cells with a light gray background, in the port matrix window of the **Edit Port Matrix** *XX* **for Switch** *YY* dialog.

You can also choose a port to which to create a dedicated connection. Note that ports with dedicated connections are not available for other connections. Ports with dedicated connections are indicated as rows of cells with a light gray background, in the port matrix window of the **Edit Port Matrix** *XX* **for Switch** *YY* dialog.

Activating a switch configuration for z/OS

Using the **Activate** button in the **Edit Port Matrix** dialog, you can activate a switch configuration after editing it.

Deleting a Switch Configuration

Select the switch configuration in the list box of the **Configurations for Switch** *XX* dialog and press the **Delete** button. HCM will display a confirmation dialog before deleting the switch.

Crossbar switches

A *crossbar* is a static switch through which you can route physical connections between controllers and processors. The crossbar switch contains a number of *channel interfaces* on its top, which can connect to objects above it, such as processors, ESCON switches or other crossbars. It also contains a number of *control unit interfaces* on its side, which can be connected to objects below it, such as controllers or other crossbars. (See Chapter 7, "Connecting objects in your configuration," on page 155 for more details).

Crossbar channel interfaces are always parallel (bus and tag). You can create, edit or delete crossbars, order their interfaces or connected objects, and define crossbar configurations.

Creating a crossbar switch

Choose **Crossbar switch...** from the **Create** menu to display the **Create Crossbar Switch** dialog. Fill in the information and press **OK**. New and unconnected crossbars appear at the extreme right in the configuration diagram. After you create the crossbar, you can edit it, or connect its channel interfaces to objects above, such as processor CHPIDs or other crossbar control unit interfaces.

Editing a crossbar switch

Open the **Edit Crossbar Switch** dialog by choosing **Crossbar switch...** from the **Edit** menu and selecting a crossbar switch from the resulting list. Note that you

can rearrange and filter the list according to your needs. Alternatively, double-click on a crossbar switch object in the configuration diagram. Use the edit dialog to change the characteristics of the crossbar or add user data. You can connect the crossbar to objects above it, edit objects below it, document crossbar configurations, or re-order crossbar interfaces.
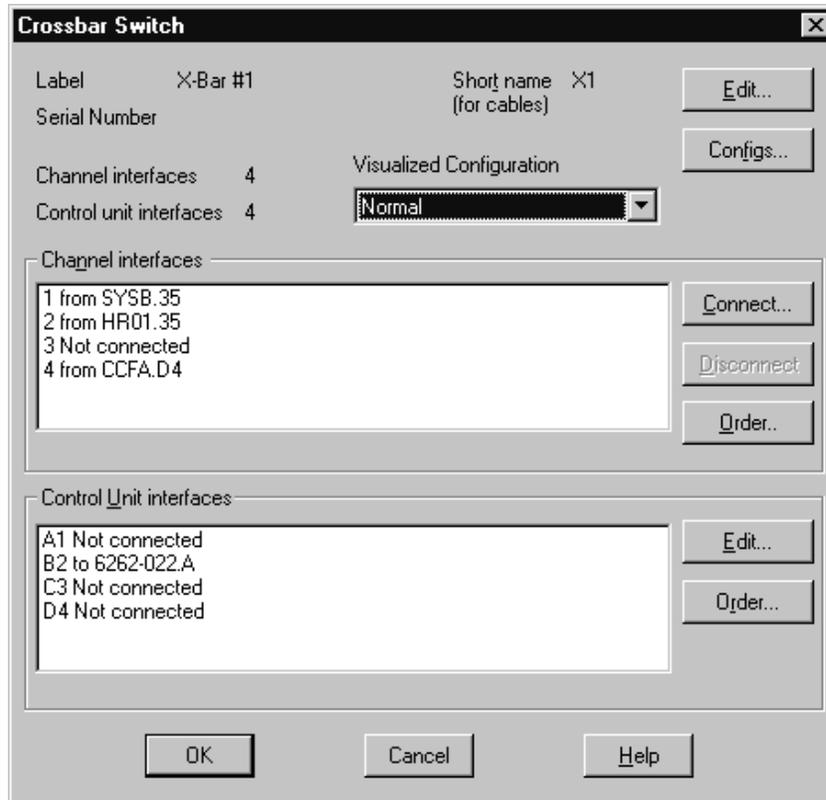
Figure 88. Edit Crossbar Switch

Adding user data to a crossbar switch

Add or edit user data by pressing the **Info...** button in the **Edit Crossbar Switch** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Recording Crossbar Configurations

You can record different settings of crossbar switch connectivity as *crossbar configurations*. These are not used in the IODF, and do not change the logical configuration of the crossbar. Click on the **Configs...** button in the **Crossbar Switch** dialog to open the **Crossbar Configurations** dialog. You can add, delete or rename crossbar configurations for the selected crossbar switch, as well as set a default configuration.

Use the crossbar configuration diagram to document a configuration. Click on an intersection point between the crossbar's channel interface and control unit interface; a large dot appears to indicate a connection. Click on a dot to remove the connection. You can move around the diagram by using the scroll bars and arrows.

After you have documented switch connectivity in the crossbar configuration diagram, you can name the configuration to record it. Click on the **Add...** button in the **Crossbar Configurations** dialog to open a standard name dialog. Type in the

name of the new crossbar configuration. Note that all crossbar configurations will
be produced in hardcopy format when you choose the Switch Reports option in
the **Print Reports** dialog. To rename the crossbar configuration, select it and click
on the **Rename...** button to open the standard name dialog.

You can select a crossbar configuration and delete it. Pressing the **Delete** button
displays a confirmation dialog before deleting the configuration. Note that the last
crossbar configuration for a crossbar switch cannot be deleted.

Use the default configuration to designate the crossbar configuration typically
used. Select a crossbar configuration and click on the **Set** button. The default will
be displayed in the **Normal/Default configuration** field.
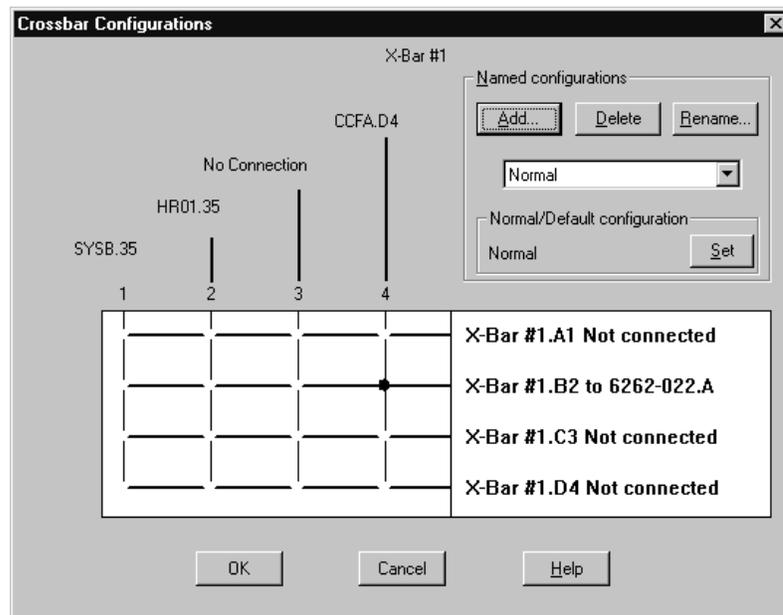
Figure 89. Crossbar Configurations Dialog

Setting the size of a crossbar switch

Open the **Edit Crossbar Switch** dialog by clicking on the **Edit...** button in the
Crossbar Switch dialog. You can change the number of channel or control unit
interfaces for the selected crossbar switch by selecting one or more values from the
appropriate drop down list boxes.

Editing controllers connected to crossbar control unit interfaces

In the **Crossbar Switch** dialog, select a control unit interface and press the **Edit...**
button in the Control Unit Interfaces section. This opens the **Edit Crossbar Control
Unit Interface** dialog which lists all controllers (and possibly other crossbars)
connected to the control unit interface. Select one and press the **Edit Object...**
button.

Ordering controllers connected to a crossbar

From the **Edit Crossbar Control Unit Interface** dialog, press the **Order...** button.
This opens an order dialog, allowing you to order the controllers (and possibly
other crossbars) that are daisy-chained on this interface.

Ordering crossbar channel or control unit interfaces

Click on the **Order...** button in either the Channel Interfaces or Control Unit
Interfaces section of the **Crossbar Switch** dialog. A standard order dialog appears,

allowing you to reorder the channel or control unit interfaces for the selected crossbar switch. See "Ordering objects" on page 61. chapter for an example of an ordering dialog.

Visualizing a crossbar switch configuration

HCM can evaluate information from crossbar switch configurations. When highlighting objects in a diagram, HCM can now use the configuration data to show what particular route is used for CHPID-to-control-unit definitions (see "Visualizing ESCON or FICON switch and crossbar switch configurations" on page 51 for more information).

To visualize a crossbar switch configuration, you first have to associate one particular configuration to each crossbar switch. Select **Visualized Matrices...** from the **View** dialog. In the following **Define Matrix Set** dialog, you can associate one configuration to each crossbar switch. Alternatively, you can select a configuration for a specific switch in the **Crossbar Switch** dialog. You can save these definitions and display them later using **Save Matrix Set** and **Restore Matrix Set** under the **View** menu.

Deleting a crossbar switch

Select the crossbar switch object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the crossbar switch.

Cabinets

Fiber management cabinets are an important part of many configurations. HCM supports the routing of fiber links through any number of patchports. See "Editing fiber links" on page 192 for more information.

In HCM, all fiber management cabinets are simply called "cabinets" regardless of how they are used. Cabinets are part of the physical information that you can add in HCM to supplement the logical information from the IODF. Each cabinet can have one or more **panels**, arranged one above the other. A cabinet has a number of places to hold panels, called **panel spaces** in HCM. Panels can be numbered starting from zero or one, from the top or the bottom of the cabinet.

A *patchport* comprises a pair of ESCON/FICON adapters or couplers. Any number of patchports can participate in a fiber link. To determine the total number of patchports in a cabinet, you must add the number of patchports of each defined panel of the cabinet.

HCM provides a predefined set of patchport naming formats. In addition, you can create your own customized naming formats. In either case this is done by providing a *Patchport Naming Format* to the cabinet definition. Individual patchport labels can be edited explicitly to overwrite the naming format in those cases where the format language is not adequate. See "Syntax of the user-defined patchport naming format" on page 151 for more information.

In the case where HCM provides a predefined set of patchport naming formats, a unique *patchport label* is created for every patchport. A patchport label is four or five characters long, in one of the following formats:

- **Consecutive numbers**. The first one or two letters are the short name of the cabinet. The following three numbers are the patchport number within the cabinet.

- **Port and group** consists of alternating letters and numbers. The first one or two letters are the short name of the cabinet. This is followed by the panel number (within the cabinet), the port group letter (within the panel), and the patchport number (within the port group). For example, the patchport label F2C4 denotes patchport 4 within port group C of panel 2 in cabinet F. Port groups are lettered from A to Z, skipping the letters I and O. Patchport numbers can start from 0 or 1.

Graphically, cabinets appear as unconnected boxes to the right of other objects to simplify the configuration diagram. However, the cabinet's patchports that form part of the fiber link between two objects are drawn as small rectangles in the connection. Selecting a patchport highlights the entire fiber link through that patchport.

You can include or exclude cabinets and patchports from your view of the configuration diagram with the appropriate **View** menu commands. After you have created cabinets and patchports, you can make any permanent connections among them, before making the logical connections in your configuration and editing the fiber links. For more information, see "Editing fiber links" on page 192 and "Infrastructure connections" on page 192.

Creating a cabinet

Choose **Cabinet...** from the **Create** menu to open the **Create Cabinet** dialog.

Enter the cabinet information and press **OK**. The following guidelines are recommended for creating a unique ID of one or two letters for the cabinet:
1. The ID is alphabetic to distinguish it from the panel number that follows.
2. If all cabinets are in the same building, use one letter for the ID.
3. If the cabinets are in different buildings, use two letters for the ID, with the first letter indicating the building.

You can suggest where HCM should draw the cabinet and its patchports in relation to other objects in the configuration diagram. Change the position of individual patchports using the **Position Patchport** dialog.

Note: You can change the number of patchports of a panel using the **Panel Attributes** dialog.

Editing a cabinet

Once the cabinet is created, you can edit cabinet information, add user data, and add or remove panels. You can also add or modify patchports (see "Patchports" on page 148). To display the **Cabinet** dialog, choose the **Cabinet...** command from the **Edit** menu. Select the desired cabinet from the list that appears. Note that you can rearrange and filter the list according to your needs.

To edit the cabinet information, click on the **Edit...** button in the **Cabinet** dialog to open the **Edit Cabinet Details** dialog. Make your changes by overtyping the fields and press **OK**. The changes appear in the **Cabinet** dialog.

If you increase the number of panel spaces, the new panel spaces are created at the bottom if they are numbered top down, or vice versa. Decreasing the number creates the opposite effect; the highest numbered panels are deleted first. Note that you can only delete a panel if none of its patchports is part of the configuration diagram.

Adding user data to a cabinet

Add or edit user data by pressing the **Info...** button in the **Cabinet** dialog to display the standard user data dialog. For more details, see "User data fields" on page 44.

Deleting a cabinet

Select the cabinet object in the configuration diagram and press the DEL key. HCM will display a confirmation dialog before deleting the cabinet.

As the user-defined formats will be deleted as well, you can save your formats (through cut & paste) in a separate file, so you can use your own created formats again.

Patchports

A *patchport* comprises a pair of ESCON/FICON adapters or couplers. Any number of patchports can participate in a fiber link. To determine the total number of patchports in a cabinet, you must add the number of patchports of each defined panel of the cabinet. HCM creates a unique *patchport label* for every patchport.

To add a panel of patchports, press the **Add...** button in the **Panels** section of the **Cabinet** dialog (Figure 90 on page 149). This will open the **Add Panels** dialog. In this dialog you can now choose the **Patchport Naming Format**. The drop-down list contains a number of predefined format strings as well as any additional user-defined format strings that are defined in currently existing cabinets in this HCM configuration file. See also "Syntax of the user-defined patchport naming format" on page 151.

To remove a defined panel of patchports, press the **Remove...** button. To change the order of panels, select a panel and press the **Move Up** or **Move Down** buttons. To change the naming format of a specific panel the patchports, select that panel and press the **Attributes...** button in the **Panels** section.

To view the patchport dialog panel, double-click on a cabinet in the configuration diagram (or choose **Cabinet...** from the **Edit** menu and double-click the appropriate cabinet). This opens the panel similar to that shown in Figure 90 on page 149.

You can view all patchports and their connections, make or break connections, position patchports in the configuration diagram, locate the currently selected patchport, or edit the fiber link of which the selected patchport is a part. Buttons are disabled when they are not valid for the selected patchport.

To change your view of the patchports in a selected panel, use the scroll bar. Display or hide the cables and objects connected to the front or back of each patchport by clicking the **Front** or **Back** check box. You can display all cables and objects connected to each patchport, or only the nearest cable and object, using the **Show Entire Link** check box.
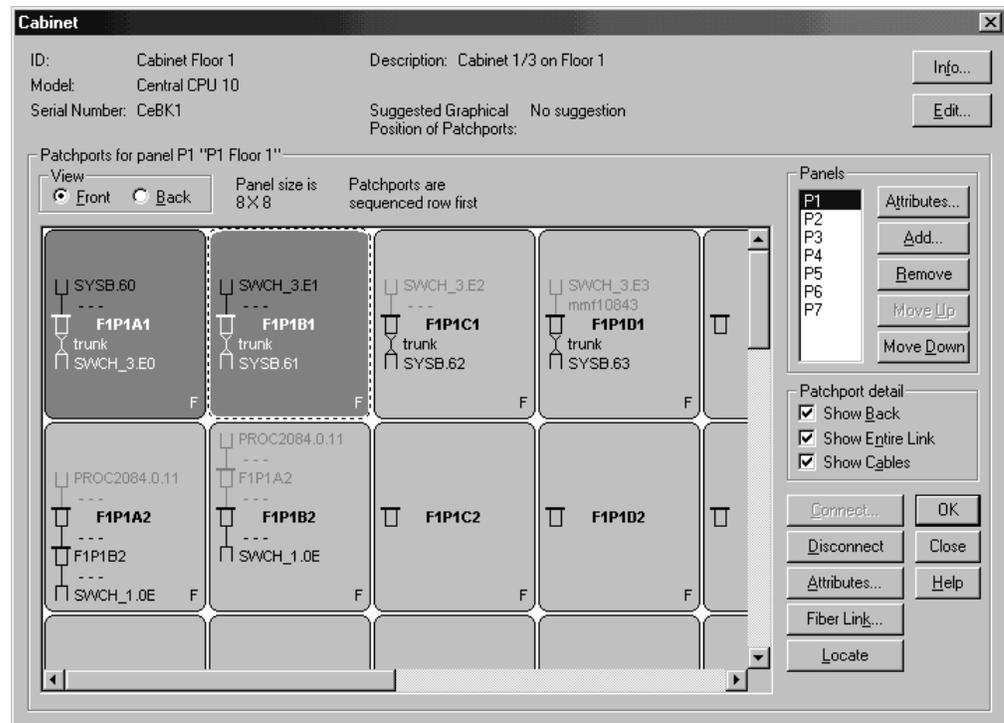
Figure 90. Cabinet Dialog

The **Cabinet** dialog offers the possibility of viewing the patchport from the front or the back. For this purpose two radio buttons have been provided on the front panel in the **View** box.

It is also possible to select a number of patchports simultaneously. This gives you the possibility of performing operations on the selected patchports as a group. Specifically, the **Connect..., Disconnect, Attributes...**, and **Locate** buttons can operate on all of the selected patchports at once.

Cabinet dialog controls

The following explains the controls on the **Cabinet** dialog:

1. **Patchports for panel X** group box.
   This large group of controls shows the attributes and patchports for the panel selected in the *Panels* list box. As the selection in the *Panels* list box changes, the label of this group box changes to display the panel ID and description. The text fields indicating panel size and the sequencing attribute are also updated. Of course, the patchport viewing area is updated, too, to show the patchports for the selected panel, but the horizontal scrolling position remains constant, as long as the new panel being displayed has enough columns. The patchport viewing area shows patchports for the selected panel and may be scrolled vertically or horizontally to show all the patchports. The patchport label is shown in bold near the centre of the patchport. Connection information for the back of the patchport is shown above the label, while connection information for the front is shown below.
   A couple of visual cues indicate what view is being shown:
   - An **F** is shown in the bottom right corner of every patchport if the front view is shown. This is changed to a **B** in the top left when the back is shown.

- To indicate the connections that are affected by **Disconnect**, connections extending from the "viewing" side are written in black, and connections extending from the other side are shown in gray.

2. **Panels** group box.
   This box shows the labels of all of the panels in the cabinet. The **Add** and **Attributes...** buttons lead to dialogs that have an editable combo box for the panel's patchport naming format. Changing the naming format in this dialog and clicking OK will update the patchport viewing area to reflect the new patchport names. This box is also used to add or remove panels and move up or down the **Panels** to change the order.

3. **Patchport detail** box.
   The check boxes here control the level of detail shown in the patchport viewing area. Connection information for the patchport's immediate neighbor is always displayed for the side being shown.

   - The **Show Back/Show Front** check box changes depending on the particular view. If it is checked, then the viewing area shows (in gray) connection information for the side not currently being viewed.

   - The **Show Entire Link** check box determines how much of the connection is displayed. If this is not checked, then only the neighbor immediately connected to the back or front of the patchport is shown. Checking this option will cause the patchport to expand so it shows not only the immediate neighbor(s), but also all the other interfaces in the complete fiber link. This option affects only the side being viewed if the **Show Back/Show Front** option is turned off.

   - The **Show Cables** check box expands the patchport further, to accommodate the labels of cables involved in the fiber link. The names of the cables (or **- - -** for anonymous cables, or **Trunk** for anonymous trunks) are shown between the interfaces they serve.

4. **Connect** button.
   This is disabled if the "viewing" side of one or more of the selected patchports is already connected. Clicking it invokes the **Select Connection Target(s)** dialog. The interface(s) that are selected from the subdialog will be connected directly to the patchports selected in this dialog. Interfaces that are already involved in complete fiber links will be valid connection targets for patchports. Connecting to such interfaces will insert the patchport and its trailing patchports into the fiber link (you will be asked to confirm this type of connection). This allows infrastructure connections to be established after a load.

5. **Disconnect** button.
   This is disabled if none of the selected patchports are connected on the "viewing" side. Clicking this button will attempt to disconnect all the selected patchports that are connected on the viewing side. Disconnect works as follows:

   - If one or more of the patchports is connected "permanently" on the viewing side, then HCM will ask the user to confirm the disconnect action. In this case, these connections will be marked as mutable before proceeding.

   - For each of the patchports:
     - If the patchport is involved in a "logically complete" connection, then HCM will remove this patchport from the chain together with any others that are permanently connected to it without breaking the logical link. If all of the connections extending from the other side of the patchport are permanent, then the disconnect action will be refused. If the disconnect

action succeeds, HCM will issue a message to indicate that the patchport chain(s) were removed without breaking the logical connection.

– If the patchport is not involved in a "logically complete" connection, then the patchport will simply be disconnected from its partner.

Note: HCM has deliberately been designed so that "logically complete" connections cannot be created or broken via the **Cabinet** dialog. The primary purpose of **Connect...** and **Disconnect** on the **Cabinet** dialog is to make (or in rare cases, break) infrastructure connections. Otherwise, use the **Switch** or **Controller** dialog to establish or break connections.

6. **Attributes...** button
Clicking this button will invoke the **Patchport Attributes** dialog panel.

7. **Fiber Link...** button
This is disabled if none of the selected patchports is involved in a complete fiber link. Clicking this button will invoke the **Edit Fiber Link** dialog (a number of times in succession if necessary). See "Editing fiber links" on page 192 for information on connecting patchports.

8. **Locate** button
This is disabled if none of the selected patchports is involved in a complete fiber link. Clicking it will close the dialog and highlight all the selected patchports that are permitted to appear in the dialog. This option is only available for patchports that are part of the configuration diagram.

Syntax of the user-defined patchport naming format

To create user-defined patchport names, you must define a format string to the cabinet definition. A format string consists of any sequence of characters that may be embedded with proxy strings. Proxy strings are delimited by '%' characters and are replaced with the appropriate text when a patchport name is displayed.

Following proxies are allowed:

%CABINET%
    ID of the cabinet

%PANEL%
    panel number

%GROUP%
    port group number

%PORT%
    patchport number

The numerical proxies (%PANEL%, %GROUP%, and %PORT%) can be qualified by width, conversion, and ordering specifications, so that the general syntax of a numerical proxy is

%[width][conversion] <proxy>{/<qualifier>}%

where [ ] indicates that the enclosed items are optional, and { } indicates that the enclosed items may appear zero or more times. The individual items are broken down as follows:

width   A decimal number in the range of 0 to 4. This indicates the width in characters of the actual text that should appear here. If the conversion is decimal or hex (see below), leading zeroes will be used to pad to the appropriate width; if the conversion is alphabetical, leading underscore

characters will be used. If the width is zero or omitted, the actual text will use only as many characters as necessary.

conversion

d   Convert number to text using decimal format

x   Convert number to text using hexadecimal format

c   Convert number to text using alphabetical format (i.e. 0 converts to A, 1 converts to B, etc.). The letters 'I' and 'O' are left out of the alphabet. Wrapping past Z will use an extra letter (e.g. AA).

If no conversion character is specified, decimal conversion is used.

<proxy>
PANEL, GROUP, or PORT

<qualifier>
Always preceded by '/'

0   Start numbering at zero (not valid for alphabetical conversion)

1   Start numbering at one (not valid for alphabetical conversion)

L   Start numbering at left (numbers increase as we move from left to right)

R   Start numbering at right (numbers increase as we move from right to left)

T   Start numbering at top (numbers increase as we move from top to bottom)

B   Start numbering at bottom (numbers increase as we move from bottom to top)

H   Numbering proceeds in horizontal direction from origin, proceeding to the next row when the end of the row is reached

V   Numbering proceeds in vertical direction from origin, proceeding to the next column when the end of the column is reached

G   Port numbering is with respect to group

P   Port or group numbering is with respect to panel

C   Port, group, or panel numbering is with respect to cabinet

Note: Those qualifiers that are grouped together above may not be used in combination; for instance %PORT/0/1% would not be valid (numbering may start from either 0 or 1, but not both).

Defaults:
The default qualifiers are ***/0/L/T/H***, indicating that ordering starts at zero from the top left and proceeds horizontally first.

As for the scope qualifier (G, P, or C), the default is
/G for proxy %PORT%, */P* for proxy %GROUP%, and
/C for proxy %PANEL%.

Regular text can be included in the format string.

Comments, delimited by curly braces, can be embedded in the format string. All
characters between the leading '{' and the trailing '}' will be ignored. Comments
are not allowed within the proxy delimiters.

The back slash '\' character can be used (outside of comments and proxies) to
allow the '%', '{', and '}' characters to be treated as regular characters.

In addition, '\\' is treated as a single back slash.

Chapter 7. Connecting objects in your configuration

This section explains how to connect objects to one another, after you have built your configuration and modified objects in the configuration diagram. Connecting involves making both physical connections between physical objects (strings, controllers, switches, machines) and logical definitions between logical objects (devices, control units, switch ports, CHPIDs, and so on).

A *connection* is a known physical path 'realized' by one or more cables. Note that purely physical objects such as cables, patchports, crossbar switches, or converters may appear in the connection between a controller and a processor, but are not taken into account by the connection. These objects are not represented in the IODF; you must add and connect them in HCM.

The *physical connection* between two objects enables appropriate *logical definitions* between these and other "reachable" objects to be made. Objects must be physically connected before they can be logically defined. When an IODF is loaded into HCM, the IODF logical definitions are represented as connections between objects in the diagram, and as attribute information in the appropriate dialogs. Use HCM to add, delete or modify physical connections and associated logical definitions.

In HCM, you connect objects that are lower on the diagram to those that are higher. For example, you connect strings to controllers (not vice versa) and you connect controllers to processors (not vice versa).

The recommended order in which you connect objects is from the top of the diagram down. For example, connect controllers (to processors) before connecting strings (to controllers), and connect switches (to processors) before connecting controllers to switches.

Objects are actually connected through their *interfaces*. A string interface for a string of I/O devices can connect to the device interface of a controller. The controller in turn can connect through its channel interface to a processor CHPID, possibly through switch ports, crossbar control unit and channel interfaces, cabinet patchports or converter interfaces.

There are two types of connections: serial and parallel. *Serial* connections (also called *fiber link*) are always point-to- point from one serial interface to another. Serial connections use optical fiber technology.

Parallel connections use "bus and tag" cables. Parallel interfaces can be *daisy chained*. A controller with a parallel interface typically has two plugs: one is connected to a processor CHPID, the other to another controller or terminator cap. Note that any devices attached to daisy chained controllers can "reach" the processor connected to the first controller in the chain. That is, all devices are physically connected and logically defined to the processor, even though only one controller is directly connected.

Controllers and crossbar switches can be daisy chained. The rule of thumb is "any interface on the top side of an object which is a parallel interface can be daisy chained to similar object interfaces". You can order daisy chained objects.

Processor CHPIDs and controller channel interfaces can be serial or parallel. ESCON and FICON switches have serial interfaces; crossbar switches have parallel. Converters (P->S or S->P) may, therefore, be required in the connection between a controller and processor.

Connecting objects

To make a typical connection, double-click on the object that you wish to connect upwards to another. An appropriate edit dialog will appear; select the object's interface and press the **Connect...** button. Another dialog appears from which you can select a target object interface to connect.

In the case of connecting devices to controllers, or controllers to processors, switches, or crossbars, you can choose to make a physical, standard or custom connection.

Physical connection:

Makes a physical connection with no logical definitions.

Standard connection:

Makes the physical connection and all appropriate logical definitions, using standard values. The standard values are determined by HCD and depend on the UIMs for the control units or devices involved, the type of CHPIDs involved, and existing connections.

Custom connection

Makes the physical connection, and allows you to customize the attributes of selected logical definitions. Note that you can customize connections between a controller and a processor, ESCON or FICON switch, or crossbar using the **Standard ...** button in the **Select Connection Target(s)** dialog. For details, see "Connecting controllers to processors and switches" on page 163.

You will typically choose to make standard connections.

HCM will check that the connections you make are valid. HCM automatically creates ESCON converters when they are required. Note that you do not have to disconnect a connection before reconnecting to another interface.

The new connection will initially have no physical cable assigned. You can assign a cable from the unused cables pool or from inventory. Cables assigned to connections which have been disconnected or reconnected become *unused cables* which "remember" where they were last used. Cables in inventory are generally available for use by any object. See Chapter 8, "Physical connections," on page 187 for more details.

Tip:

Note that making or breaking a connection may move the selected object such that it is no longer visible in the current view. You may have to zoom out and then in, or scroll around to find the selected object.

Positioning objects along daisy-chained connections

A number of controllers can be daisy-chained on a CHPID. Daisy-chains can also exist under crossbar switches, S->P converters, and controllers.

If you are connecting an object to a target that is already connected to one or more objects, the standard **Position Interface** dialog will appear to allow you to *position* the object you are connecting with respect to the others.
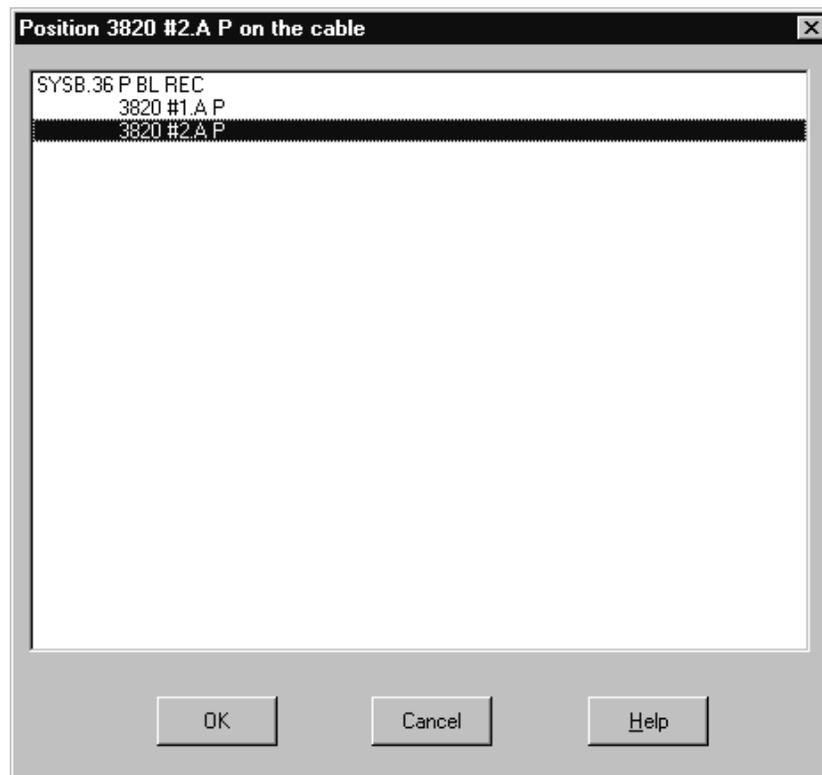
Figure 91. Positioning an Object for Daisy-Chained Connections

Select an object in the list box and drag it through the list to the object that will immediately follow it. Release the mouse button; the object will be repositioned in the daisy- chain. Alternatively, select the object and use the arrow keys on the keyboard to move it up or down the list. By default, HCM attaches the new object to the end of the chain.

Connecting devices to controllers

I/O devices connect to processors through controllers. If the controller is daisy chained to other controllers, all devices attached to controllers in the chain can "reach" any connected processors.

More specifically, the *string interface* on a string of devices connects to the *device interface* on a controller; the controller channel interface in turn eventually connects to one or more processor CHPIDs. The connection between device and controller is always between parallel (bus and tag) interfaces, whereas the connection between controller and processor may be parallel and/or serial.

In some cases the I/O devices may be *integrated* into the controller.

In the **Select Controller Device Interface** dialog, there are three types of connection. You can make a purely *physical connection* with no logical definitions. Or you can make a *standard connection* with appropriate logical definitions, from the device to all "reachable" processors through the selected control unit. In this

most common type of connection, the device is defined to all control units in the controllers, any connected processors, and all partitions in any connected processors. Alternatively, you can customize the logical definition attributes yourself by making a *custom connection*, first selecting the specific device and control unit for which definitions will be made.

Creating a string under a controller

Double-clicking on a controller in the configuration diagram opens the **Controller** dialog. Alternatively, choose **Controller...** from the **Edit** menu, and select the desired controller from the list which appears. Select the **Create...** button in the Strings/Devices section to display the **Create String** dialog. Fill in the information and press **OK**. Strings and their I/O devices created from the **Controller** dialog are not automatically connected and defined to the selected controller. Instead, the **String** dialog appears, where you can choose how to connect the string's interfaces.

If there is more than one physical description possible for the string, pressing **OK** will first display the **Choose Physical Description** dialog before the connection dialog.

Selecting an existing string under a controller

Double-click on the desired controller to open the **Controller** dialog. The Strings/Devices section in the dialog lists any strings/devices connected to the controller through its device interfaces. If I/O devices have no physical string (they are integrated into the controller), the "virtual strings" containing them are displayed.

Select a string/device and press the **Edit...** button to display the **String** dialog. You can then open the **Select Controller Device Interface** dialog by selecting one or more string interfaces and pressing the **Connect...** button.

Note that you can also open the **String** dialog by choosing **String...** from the **Edit** menu, or by double-clicking on a string in the configuration diagram.
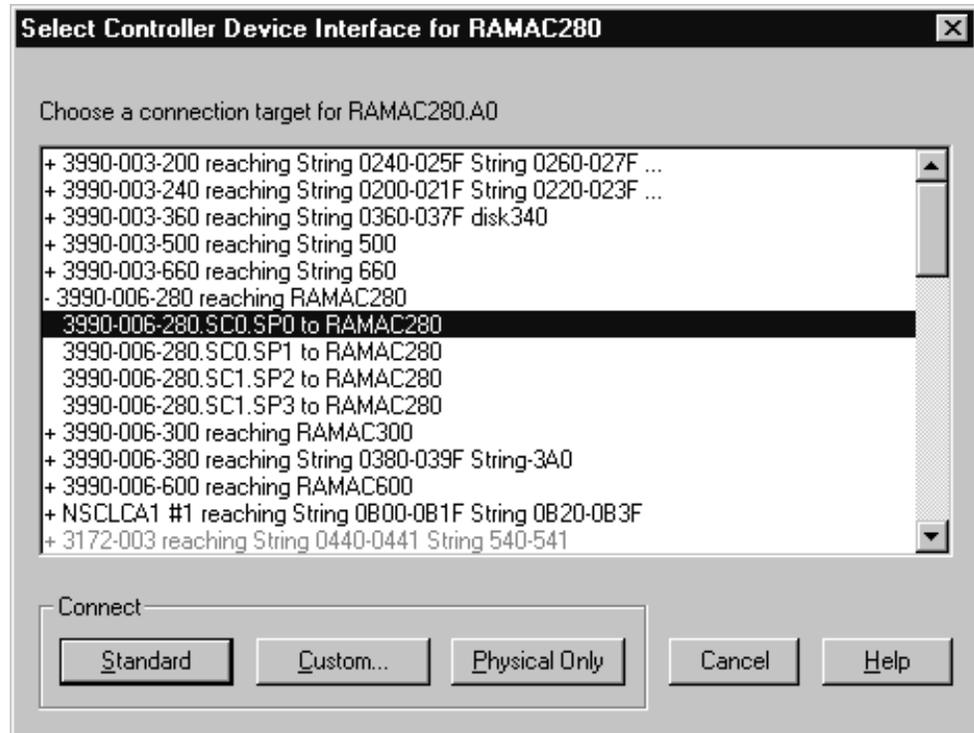
Figure 92. Select Controller Device Interface Dialog

Connecting the string to a controller device interface

Once you have displayed the **Select Controller Device Interface** dialog (see
"Selecting an existing string under a controller" on page 158), you can select a
controller device interface as the target to which you wish to connect the selected
string interface(s).

The list box lists all device interfaces on controllers that are available to connect to
the selected string interface(s). Double-click on a controller with a plus (+) sign to
expand it to display all possible device interfaces for selection. Double-click on a
controller with a minus (-) sign to contract it.

Invalid targets (device interfaces or controllers containing device interfaces which
cannot reach any valid control units) are disabled. The validity of control units for
connection is determined from HCD.

You can also choose which type of connection to make: physical, standard, or
custom.

Making a physical connection

Select a controller device interface from the list box in the **Select Controller Device
Interface** dialog and press the **Physical Only** button. This option will connect the
selected devices to the controller. However, no logical definitions are created
between the devices and all processors connected to the controller.

Making a standard connection

Select a controller device interface from the list box in the **Select Controller Device
Interface** dialog and press the **Standard** button. If there is only one control unit in

the segment of the selected controller, this option will connect the selected devices to the controller. HCM will automatically create the necessary logical definitions between the devices and all processors connected to the controller, using standard values. The standard values are determined by HCD and depend on the UIMs for the control units or devices involved, the type of CHPIDs involved, and existing connections. If standard values cannot be determined, HCM proceeds as if the **Custom...** button was pressed.

If there is more than one control unit in the segment of the selected controller, the **Reachable Control Units** dialog appears to allow you to specify the control unit(s) to which to connect. Add the physically reachable control units that you wish to include in the connection to the Include list and press **OK**. HCM makes the connection and all appropriate logical definitions.
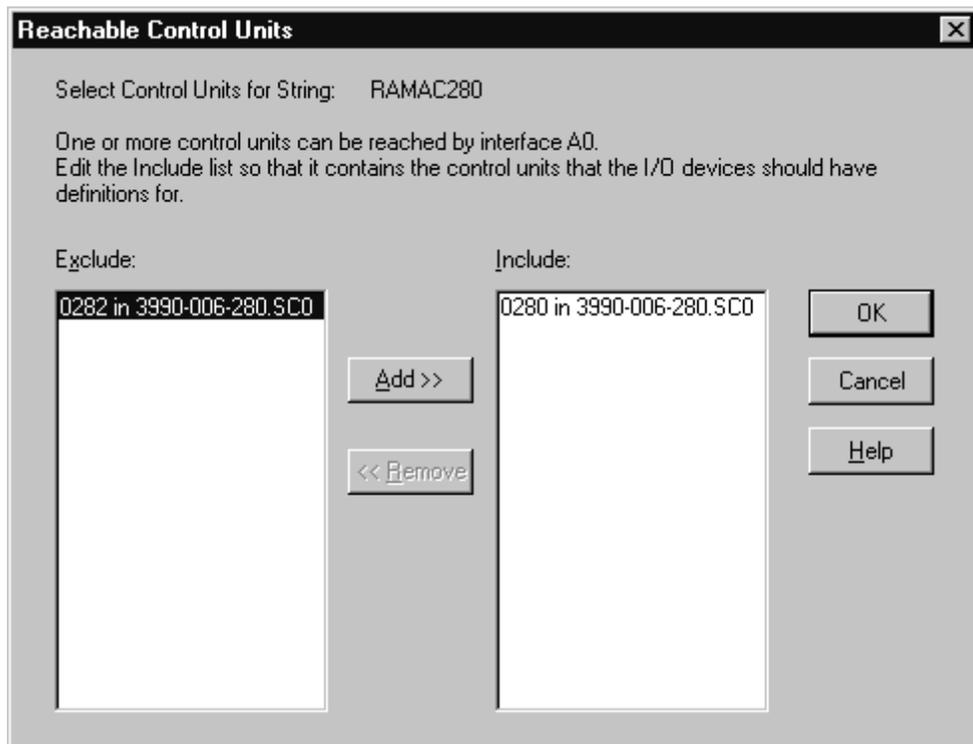
Figure 93. Reachable Control Units Dialog

Making a custom connection

Select a controller device interface from the list box in the **Select Controller Device Interface** dialog and press the **Custom...** button. This option will display the **Affected I/O Devices** dialog to allow you to select one or more devices for which to make the device <=> processor definitions.

Note that you can view control unit numbers, serial numbers, VOLSERs, descriptions, etc., for the devices in the Affected I/O Devices list box depending on the selection made in the **Columns** dialog.
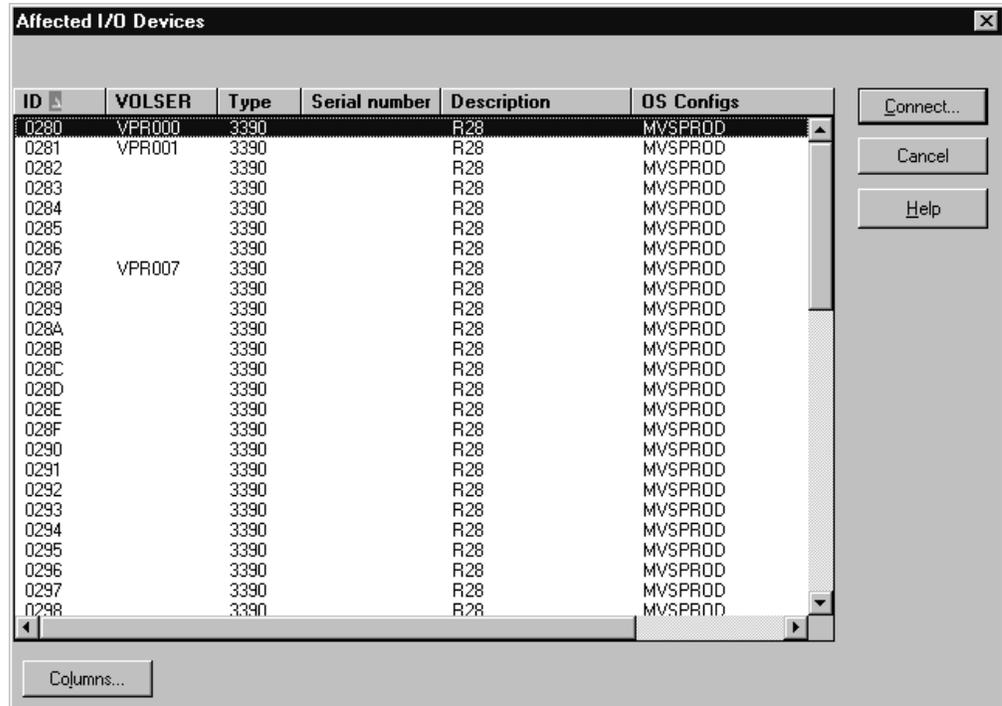
Figure 94. Affected I/O Devices Dialog

After you selected one or more devices, press the **Connect...** button. If there is more than one control unit in the controller segment, the **Reachable Control Units** dialog opens. Add the physically reachable control units that you wish to include in the connection to the Include list and press **OK**. The **I/O Device <=> Processor Definition** dialog appears. Otherwise, HCM opens the **I/O Device <=> Processor Definition** dialog directly.
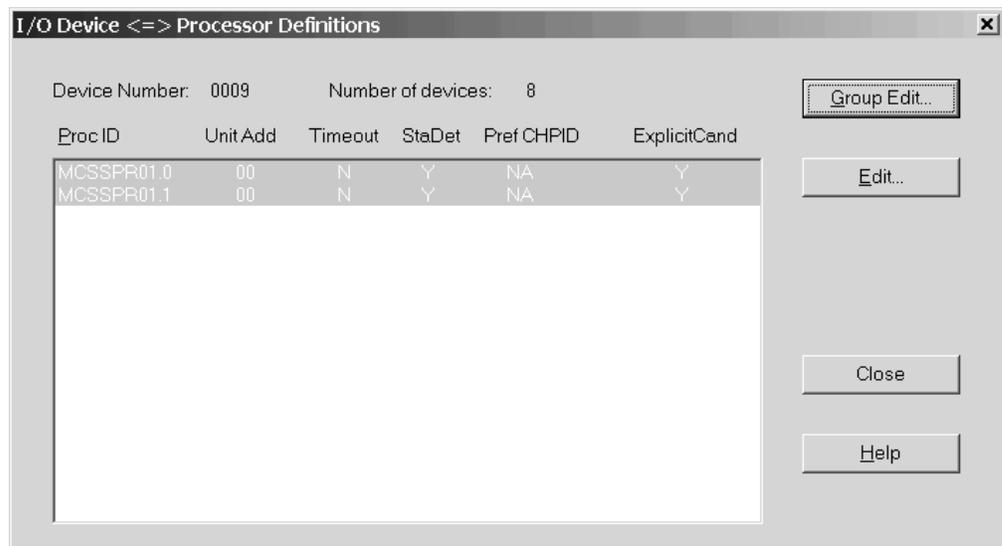
Figure 95. I/O Device <=> Processor Definitions Dialog

The **I/O Device <=> Processor Definition** dialog lists the parameters for the connection between this control unit and each processor to which it is defined. Select one or more processors and press the **Edit...** or **Group Edit...** button. The

Edit I/O Device <=> Processor Definition dialog appears which lets you edit the specified logical definition. If you pressed **Edit...**, this dialog will appear once for each selected processor. If you pressed **Group Edit...**, the dialog appears only once and any changes you make affect all the selected processors.
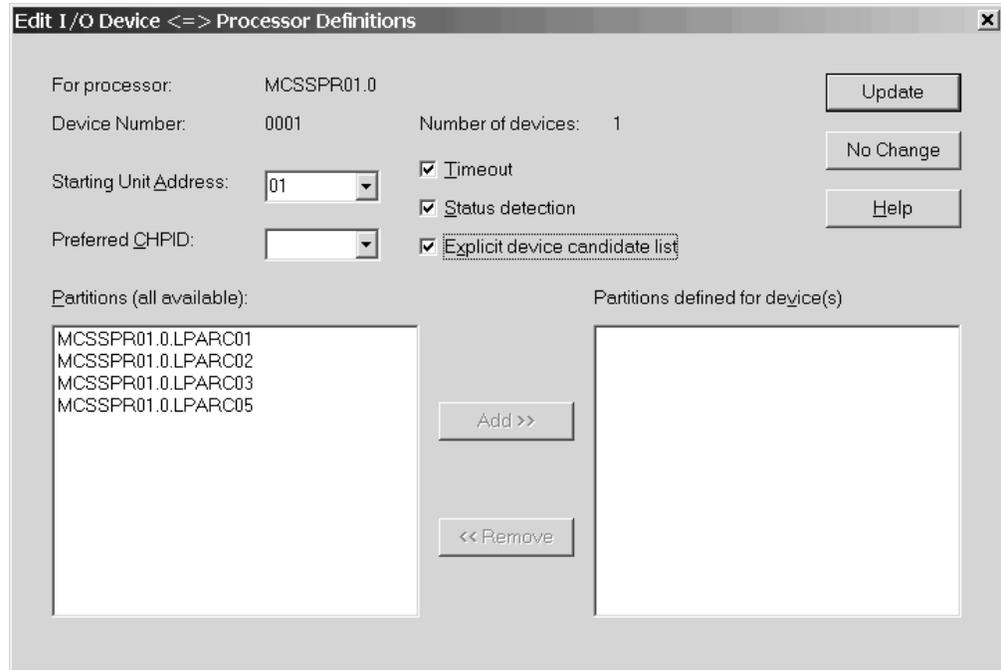
Figure 96. Edit I/O Device <=> Processor Definitions Dialog

In this dialog, you can modify the attributes of the logical definition for the selected device(s), control unit and processor(s). Press the **Update** button to save your changes and close the dialog. Press the **No Change** button to close the dialog without making any changes to the device <=> processor definition.

Tip:

There is an alternative way to access the **Edit I/O Device <=> Processor Definitions** dialog for an existing device <=> processor link that does not involve making a connection between a device and controller. Double-click on a string to open the **String** dialog. Press **I/O Devices...** to open the **I/O Device List** dialog. Select a device(s) and press the **Dev<=>Proc...** button. The **I/O Device <=> Processor Definition** dialog appears; you can then select a processor(s) and press **Edit...** or **Group Edit...** to open the **Edit I/O Device <=> Processor Definition** dialog.

Explicit device and null device candidate lists

In the **Edit I/O Device <=> Processor Definitions** dialog from Figure 96, you can check the *Explicit device candidate list* box. This selection fills the *Partitions (all available)* box with a list of partitions that are reachable for the current device. Adding partitions to the *Partitions defined for device(s)* box will create the explicit device candidate list for the current device. Only the partitions contained in this list may access the device.

For XMP processors, if devices are connected to a control unit which is shared between multiple channel subsystems, some (not all) of these devices may specify an explicit null device candidate list for one or more CSSs. If you define a null device candidate list of a device for a certain CSS, then no partition of this CSS may have access to the device. Specifying a null device candidate list in HCM means to check the *Explicit device candidate list* box **without** adding candidates to the *Partitions defined for device(s)* box.

Disconnecting the string interface from a device interface

Open the **Strings** dialog by double-clicking on a string in the configuration diagram, or by choosing **String...** from the **Edit** menu. Alternatively, double-click on a controller to open the **Controller** dialog, then select a string from the String/Devices section and click on **Edit...** to open the **String** dialog. Select the connected string interface(s) from the Interfaces list box and press the **Disconnect** button. HCM will ask you for confirmation before disconnecting.

A *disconnection* involves physically disconnecting the string interface from the controller and deleting any logical definitions between the devices and any control units and processors that can no longer be physically reached.

Connecting controllers to processors and switches

Controllers connect devices to processors. Controllers can be daisy chained, making all connected processors physically reachable to all attached devices.

More specifically, the *channel interface* on the top of a controller can connect to a CHPID on a processor. This connection may be direct, or it may be through ESCON or FICON switches, crossbar switches, cabinet patchports, converters, or daisy chained controllers. The connection between controller and processor may be parallel (bus & tag) and/or serial. Converters are automatically added as necessary.

Connections between serial channel interfaces on controllers and serial CHPIDs form *fiber links*. Initially, the fiber link comprises a direct connection, but it can also include patchports, trunks, and jumper cables. See "Editing fiber links" on page 192 for more details.

In the **Select Connection Target(s)** dialog, you can ask HCM to make a *physical connection* or a *standard connection*, where you can also adjust several parameters to your needs. The physical connection is a simple physical connection, with no logical definitions. The standard connection creates logical definitions between the controllers and reachable processors. This is the most common type of connection: all control units are defined to the CHPID and processor, and all devices connected to the controller are defined to the processor and all its partitions. Depending on whether you connect a controller directly to a CHPID or to a switch, the subsequent dialogs may differ. In the first case, the **Edit Control Unit <=> Processor Definitions** dialog might appear, where you can specify some parameters in order to customize selected logical definitions. Note that default parameters are always provided in this dialog, which can then be modified. In the second case, that is, when you connect a controller with a port of a switch, the **Processor and CHPID Connections** dialog appears. There, you can customize your connections by pressing the **Attributes ...** button. Again, the **Edit Control Unit <=> Processor Definitions** dialog then appears, where you can adjust the default parameters to your needs.

Selecting a channel interface for a controller

Open the **Controller** dialog by choosing **Controller...** from the **Edit** menu and selecting a controller from the resulting list, or by double-clicking on the controller in the configuration diagram. Select an interface from the controller and press the **Connect...** button to open the **Select Connection Target(s)** dialog. This dialog allows you to connect the controller channel interface to a target interface, such as a processor CHPID, switch port or crossbar control unit interface. The dialog leads to others which allow you to create any necessary logical definitions for the physical connection.

Note: You can also select multiple controller interfaces before pressing the **Connect...** button. This opens the **Select Multiple Connection Target(s)** dialog which allows you to establish several connections in one step. For information on how to use this dialog, refer to "Connecting switches to processors or other switches" on page 172.

Connecting the channel interface to a CHPID

Connecting a channel interface to a CHPID creates a *physical route* between control units in the controller and the processor. If there is an ESCON or FICON switch or crossbar switch in the connection, the control units may access more than one processor. See "Connecting crossbar switches" on page 198 and "Connecting switches to processors or other switches" on page 172 for details.

> **Note:**
> Although the rest of this section deals with connecting the controller channel interface to a processor CHPID, you can connect the controller to a crossbar switch or an ESCON or FICON switch which eventually leads to the processor CHPID, using the **Select Connection Target(s)** dialog.
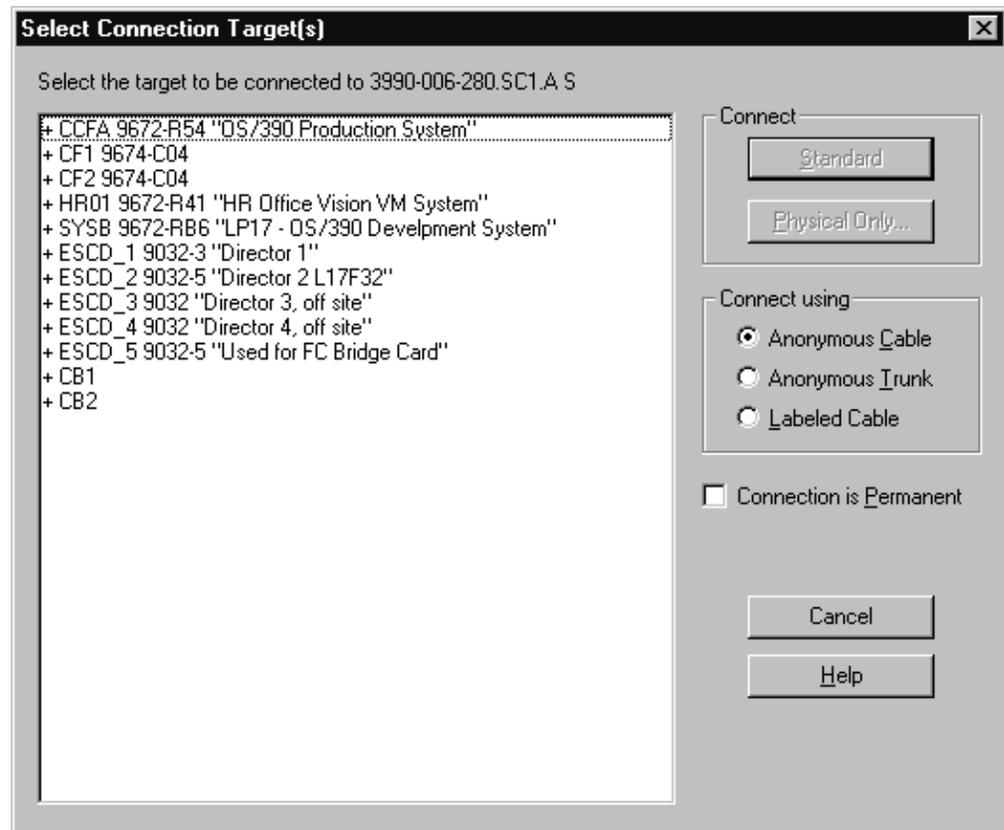
Figure 97. Select Connection Target(s) Dialog

Open the **Select Connection Target(s)** dialog by selecting a channel interface and pressing the **Connect...** button in the **Controller** dialog.

Note: If the selected controller is part of a switch, and you accessed the **Controller** dialog from the **Edit Switch** dialog, pressing the **Connect...** button displays the dialog that would appear if you had already pressed the **Standard** button in the **Select Connection Target(s)** dialog. (That is, the **Select Connection Target(s)** dialog is skipped.)

Select a processor CHPID as the target interface to which to connect. The Connection target list box lists all physical connection targets to which you can connect. Double-click on a target object with a plus (+) sign to expand it to display all possible target interfaces. Double-click on a target with a minus (-) sign to contract it. Note that double-clicking on a target interface triggers the **Standard** connect button. Invalid targets are disabled.

If you are connecting the channel interface to a CHPID that is already connected to another controller, pressing **Standard...** or **Physical Only...** will first display the **Position Interface** dialog. You can position the channel interfaces on the target object before connecting. See "Positioning objects along daisy-chained connections" on page 156 for more information on the **Position Interface** dialog.

After selecting the target CHPID, you choose which type of connection to make: physical or standard (customization of the connection is possible here). Depending on whether a connection should be established directly to a CHPID or to a switch port, further user interaction differs. Let's consider first that a connection should be established between a controller and a port of a switch. Depending on the user

interaction on the **Processor and CHPID Connections** dialog, HCM uses standard values for the attributes of the connection or the ones that the user has specified in the **Edit Control Unit <=> Processor Definitions** dialog, which appears after pressing the **Attributes...** button in the **Processor and CHPID Connections** dialog. The standard values for the attributes are determined by HCD and depend on the UIMs for the control units or devices involved, the type of CHPID involved, and existing connections. In the second case, where a connection between a controller and a CHPID is considered, HCM allows you to customize the attributes of selected logical definitions via the **Edit Control Unit <=> Processor Definitions** dialog.

Making a physical connection

Select a CHPID from the Connection target list box and press the **Physical Only...** button in the **Select Connection Target(s)** dialog. This option will connect the selected controller to the processor. However, no logical definitions are created between the controller and processor. Note that this will cause the connection to be colored as a logical/physical inconsistency on the diagram. See "Receiving visual warnings" on page 57 for more information.

Making a standard connection

Select a CHPID from the Connection target list box and press the **Standard ...** button in the **Select Connection Target(s)** dialog. This option makes the physical connection and all appropriate logical definitions. Depending on whether you want to establish a connection between a controller and a CHPID or a port of a switch, this option will take you through a series of dialogs, depending on certain conditions, to allow you to edit control unit <=> processor logical definitions.

Considering that a connection between a controller and a port of a CHPID is planned, the following dialogs might appear:

- **Affected Control Units**
- **Processor and CHPID Connections**
- **Position Interface**
- **Control Unit <=> Processor Definitions**
- **Edit Control Unit <=> Processor Definitions** dialog

In the simplest case of connecting a controller with one control unit with one channel interface to a CHPID that has no physical connections, clicking **Standard...** might immediately establish the connection without further user interaction.
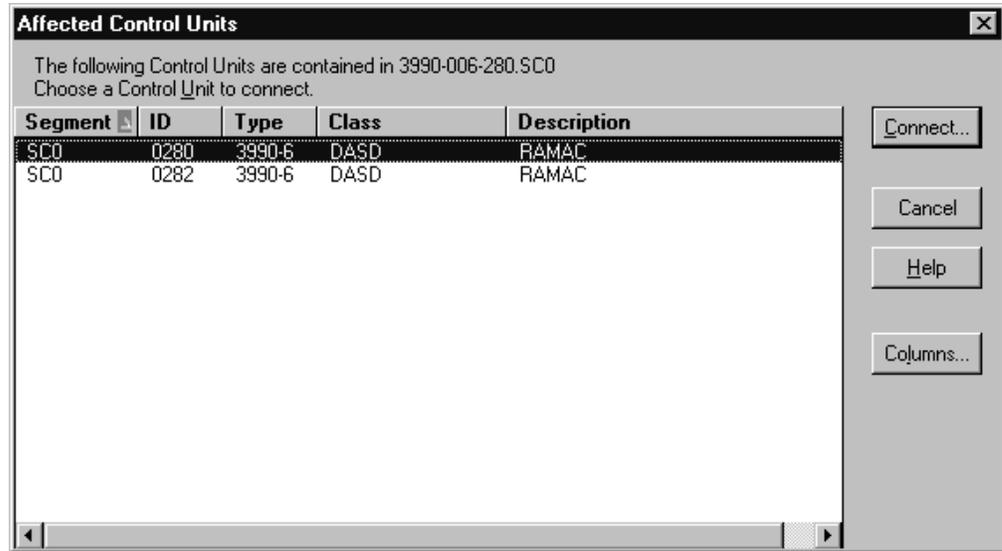
Figure 98. Affected Control Units Dialog

The **Affected Control Units** dialog appears if the selected controller channel interface can be reached by more than one control unit. Select a single control unit to define to the processor(s) and press the **Connect...** button.
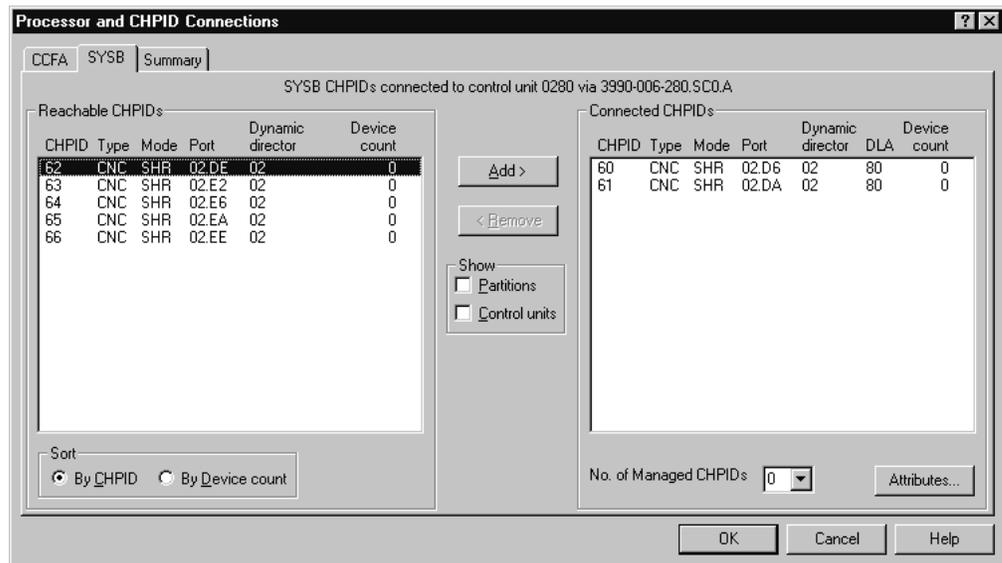
Figure 99. Processor and CHPID Connections Dialog

The **Processor and CHPID Connections** dialog appears if you connect the control unit via a switch. You can select the available channel paths of a processor by clicking on the appropriate processor tab in the dialog. The summary tab in this dialog gives an overview of which CHPIDs the connection will be made to. The interface you are connecting is shown on each individual tab. Selecting the **Show partitions** check box displays additional lines of text showing access and candidate lists for the CHPIDs in each list box entry. Selecting the **Show control units** check box displays an additional line of text for each attached control unit below the CHPID. This line will show the control unit number, its type and attached device ranges with their types.

If more than one CHPID can be reached, you can then include or exclude certain CHPIDs in the connection and press **OK** before specifying logical definition attributes.

Besides static connections, a control unit can also have connections to manageable channel paths. In order to indicate the maximum number of manageable channel paths allowed to connect between the processor to which a connection is to be established and the control unit of interest, enter the correct number in the respective entry field on this dialog. The usual configuration rules apply.

If a spanned CHPID is connected to a control unit, for example via a switch or a controller interface, then the **Processor and CHPID Connections** dialog offers tabs for each channel subsystem over which the CHPID is spanned, as shown in Figure 100.

If you connect a control unit to an FC CHPID via cascaded FICON directors, and if all involved objects can handle two-byte link addresses, then a two-byte link address might be created in case a switch address is provided for the switch to which the control unit gets connected. The following list contains all possible cases when HCM will create a two-byte link address:

(1)   If there are no connections on the CHPID, and both the processor and the switch support cascading, and the switch has a switch address, then a two-byte link address is used per default.

(2)   If there are connections on the CHPID, and the first one uses a two-byte link address, then the new connection must use a two-byte link address.

(3)   If the CHPID does not connect to the same switch as the controller interface used for the connection, then HCM must use a two-byte link address.

In all other cases, HCM will create a one-byte link address whenever there is the need to decide between a one-byte link address and a two-byte link address. For ESCON connections running over ESCON switches, two-byte link addresses are not eligible.
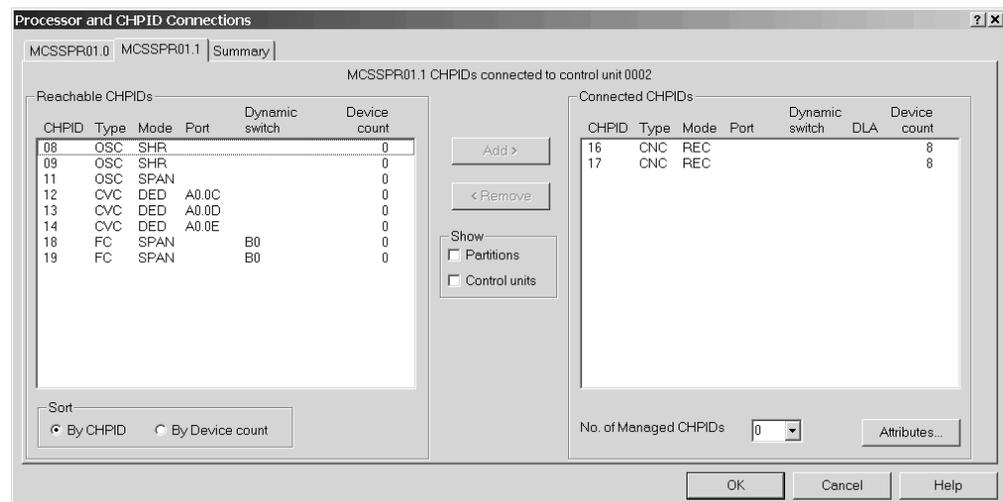
Figure 100. Processor and CHPID Connections Dialog

The **Position Interface** dialog appears if the selected CHPID is already connected to another controller (daisy chaining). This dialog allows you to position the channel interface on the daisy chain. See "Positioning objects along daisy-chained connections" on page 156 for more information.
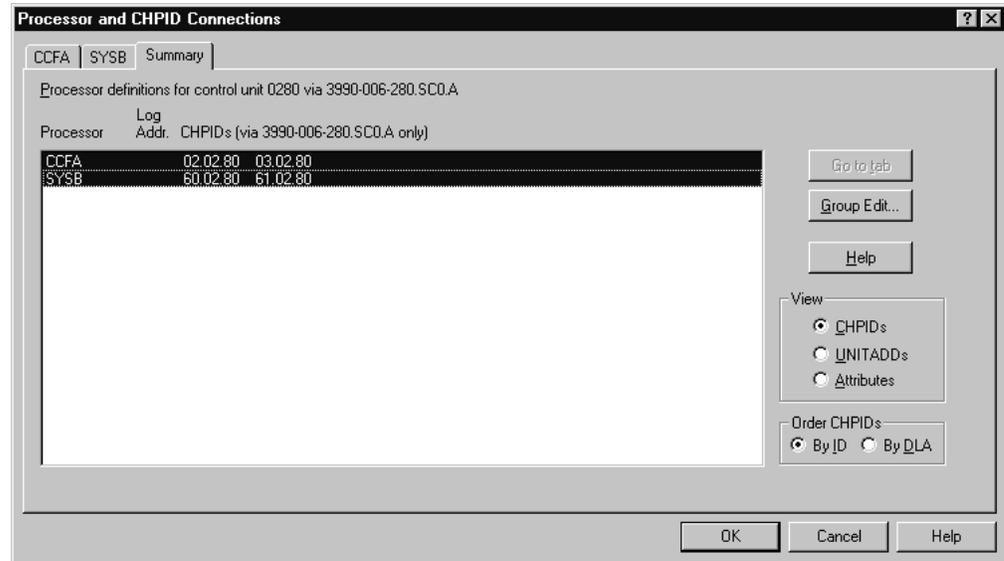
Figure 101. Processor and CHPID Connections Dialog

If you click on the Summary tab of the **Processor and CHPID Connections** dialog, you obtain the processor definitions for the control unit under focus. You can then specify the processor(s) for which you will edit the control unit <=> processor logical definitions and press the **Group Edit ...** button.

Ultimately, the **Edit Control Unit <=> Processor Definitions** dialog appears allowing you to edit the control unit <=> processor logical definitions for the selected control unit and one or more selected processors. This dialog appears also for selected logical definitions when a connection has to be established directly between a controller and a CHPID without going through a switch.

An asterisk '*' on the Summary tab of the **Processor and CHPID Connections** dialog indicates that the control unit of interest has connections defined to manageable channel paths of the indicated processor(s). This means that the control unit might be reachable by more than the statically defined channel paths when the system is up and running.

Editing logical definitions between controller and processor
Edit the attributes for the control unit <=> processor logical definition. Press the **Update** button to validate the changes and return to the calling dialog. Alternatively, pressing the **No Changes** button will allow you to leave the dialog without creating all the required definitions. This may produce a physical-logical inconsistency.

Edit Control Unit <=> Processor Definitions

Control Unit number: 0280

Control Unit type: 3990-6

Processor ID: CCFA

OK
Cancel
Help

Unit Address: 00

Unitadds...

No. of Units: 256   (0 additional ranges)

Logical address:
(Same as CUADD)

Protocol: S   (D,S or S4)

I/O Concurrency: 2   (1,2 or 3)
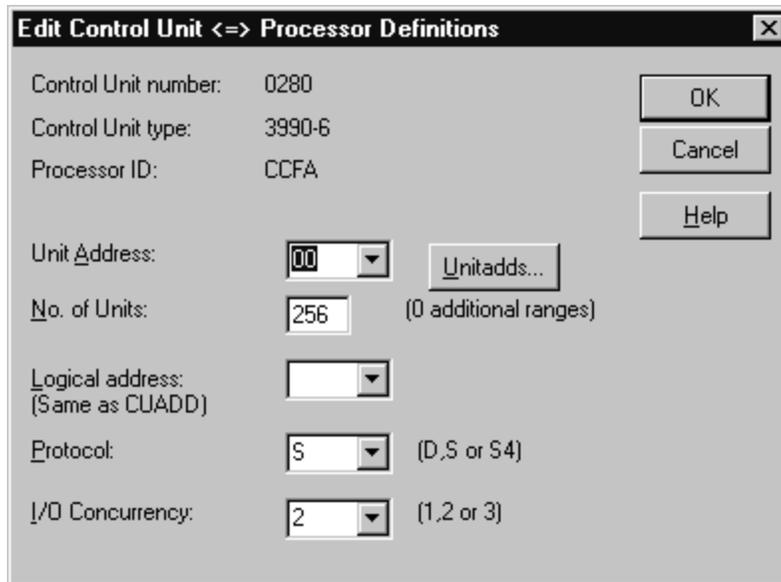
Figure 102. Edit Control Unit <=> Processor Definitions Dialog

You can specify or edit unit addresses and ranges when more than one range exists
for the selected control unit. Click on the **Unitadds...** button to open the **Unit
Address Ranges** dialog.

If you have more than one unit address range, and if your selection for **Unit
Address** and **No. of Units** comprises units from different ranges, then a
renumbering of unit addresses will occur. Always make sure that you select correct
unit address ranges for making updates.

> **Tip:**
>
> There is an alternative way to access the **Edit Control Unit <=> Processor
> Definition** dialog for an existing control unit <=> processor definition that
> does not involve making a connection between a controller and processor.
> Double-click on a controller to open the **Controller** dialog. Press **Control
> Units...** to open the **Control Unit List** dialog. Select a control unit and press
> the **Connections...** button. The **Processor and CHPID Connections** dialog
> appears on which you can select the tab of that processor or channel
> subsystem for which you are interested in the control unit <=> processor
> definition. Pressing the **Attributes...** button opens the **Edit Control Unit <=>
> Prodcessor Definitions** dialog.

Disconnecting the channel interface from a CHPID

Open the **Controller** dialog by double-clicking on a controller in the configuration
diagram, or by choosing **Controller...** from the **Edit** menu. Select the connected
channel interface(s) from the Interfaces list box and press the **Disconnect** button.
HCM will ask you for confirmation before disconnecting.

A *disconnection* involves logically disconnecting any control units from any CHPIDs
that were only reachable via the controller channel interface, as well as physically
disconnecting.

Connecting controllers to controllers (PPRC connections)

Controllers can be connected to other controllers either directly via a point-to-point connection, or via one or more switches. Such connections are used for PPRC connections, and they can be defined and documented with HCM. Creating a PPRC connection in HCM consists of two steps:

1. The physical connections need to be established. Therefore, connect the channel interfaces either via a point-to-point connection or via one or more switches with each other.

2. Create a PPRC connection consisting of up to 8 PPRC links using HCM's PPRC utility reachable via the **PPRC Connections ...** menu item in the **Edit** menu.

"Scenario V – Establish a PPRC connection consisting of several PPRC links" on page 307 runs a scenario in which a PPRC connection is set up in detail, consisting of point-to-point and switched links.

Physically connecting a channel interface to a channel interface of another controller

Before you can create a PPRC connection in HCM, the physical connections must be available to HCD. Connecting a channel interface of a controller directly to a second channel interface creates a *physical route* between two controllers. After defining such a connection, no logic connection has been established. When you define connections between channel interfaces and switches, you also establish the physical environment for the PPRC connection regardless whether one or more switches are involved.

First, consider the case of establishing a point-to-point connection via two channel interfaces. The process of connecting two channel interfaces with each other is similar to connecting a channel interface to a CHPID (see "Connecting the channel interface to a CHPID" on page 164). In the **Controller** dialog, select a controller interface and press the **Connect ...** button. In the following **Select Connection Target(s)** dialog, select an interface of one of the controllers in your configuration to which you want to establish a point-to-point connection, and press the **Physical only ...** button. Note that you can only establish a physical connection, but no logic connection, when connecting two channel interfaces with each other, hence, the **Standard** button is disabled when you select a channel interface in the **Select Connection Target(s)** dialog.

Consider now the case of a chained or cascaded connection between two channel interfaces. Again, this process is similar to connecting a channel interface to a CHPID. Here, you have to connect the involved controller interfaces with ports of a switch. For details, please refer to "Connecting the channel interface to a CHPID" on page 164. In case of a chained switch environment, you also have to connect the switches with each other. For details, please refer to "Connecting switches to processors or other switches" on page 172.

Creating a PPRC connection

After having established physical connections between the involved objects of a PPRC connection, you can create the PPRC connection itself. Open the **PPRC Connections** dialog via the **Edit** menu and press the **Create** button. This will lead you to the first page of the **Create PPRC Connections** dialog in which you can select the primary (source) subsystem. Press **Next** to open the second page of this utility on which you can select the secondary (target) subsystem to be connected to the primary subsystem. After pressing the **Next** button again, you are prompted on

the third page of the utility to choose the interfaces used to connect the selected subsystems. Here, you can specify up to eight single PPRC links. When you are finished selecting the interfaces, press **Next** once more to arrive at the last page of the utility. Here, you can enter the remaining parameters for each PPRC link. When you have specified all needed values, press the **Finish** button to complete your PPRC connection.

Editing a PPRC connection

You can edit existing PPRC connections and modify them accordingly. Open the **PPRC Connections** dialog via the **Edit** menu. Select a PPRC connection listed in the dialog, and press the **Edit** button which leads you to the first page of the **Edit PPRC Connections** utility. Here you can select which interfaces of the involved controllers should participate in a PPRC connection. After you have selected the interfaces under focus, press the **Next** button which leads you to the second page of the **Edit PPRC Connections** utility. Here, you can adjust the remaining parameters for each PPRC link. When you have specified all needed values, press the **Finish** button to complete your PPRC connection.

Importing and exporting PPRC connections

You can edit existing PPRC connections and modify them accordingly. Open the **PPRC Connections** dialog via the **Edit** menu. Select a PPRC connection in the dialog, and press the **Import/Export ...** button. This will lead you to the **Import/Export PPRC Connections** utility. On the first page of that utility, you can select whether to export or to import PPRC connections. After pressing the **Next** button, you can select whether you want to process CESTPATH commands or GEOPARM control statements. On the third page, you can specify a filename for import and export, depending on which action you have selected on page one of the wizard.

Deleting a PPRC connection

You can delete existing PPRC connections by opening the PPRC Connections dialog via the Edit menu. Select a PPRC connection listed in the dialog, and press the Delete button. A PPRC link is also deleted when a controller interface that participates in that particular PPRC link is disconnected either from the target controller interface in case of a point-to-point PPRC link, or from the switch port in the chained case.

Disconnecting a channel interface from a channel interface

Disconnecting a channel interface from a channel interface works exactly the same way as disconnecting the channel interface from a CHPID, see "Disconnecting the channel interface from a CHPID" on page 170. After having disconnected a controller interface from another controller interface or from a switch port, PPRC links in which that controller interface was involved are deleted.

Connecting switches to processors or other switches

You can insert a switch into the connection between controllers and processors. The switch *port* can connect upwards to either a CHPID or another switch port, using the **Switch** dialog and the **Select Connection Target(s)** dialog or the **Select Multiple Connection Target(s)** dialog. When you connect controllers to an ESCON or FICON switch, you also use the **Select Connection Target(s)/Select Multiple Connection Target(s)** dialog. If you connect the controller to a switch which is already connected to several CHPIDs, the controller (and any already connected I/O devices) can establish communication with all reachable processors.

Note: If you want to connect channel interfaces of a crossbar switch to an ESCON or FICON switch or to CHPIDs, you can select only one channel interface at a time and therefore you will reach the **Select Connection Target(s)** dialog only, but not the **Select Multiple Connection Target(s)** dialog.

ESCON and FICON switch ports are serial interfaces, unlike crossbar switch interfaces which are parallel (bus and tag). Switches cannot connect upwards to crossbar switches, but crossbars can connect upwards to switches (see "Connecting crossbar switches" on page 198 for more details).

To connect a switch, open the **Switch** dialog by double-clicking on a switch in the configuration diagram. Alternatively, choose **Switch** from the **Edit** menu and select the switch from the resulting list.
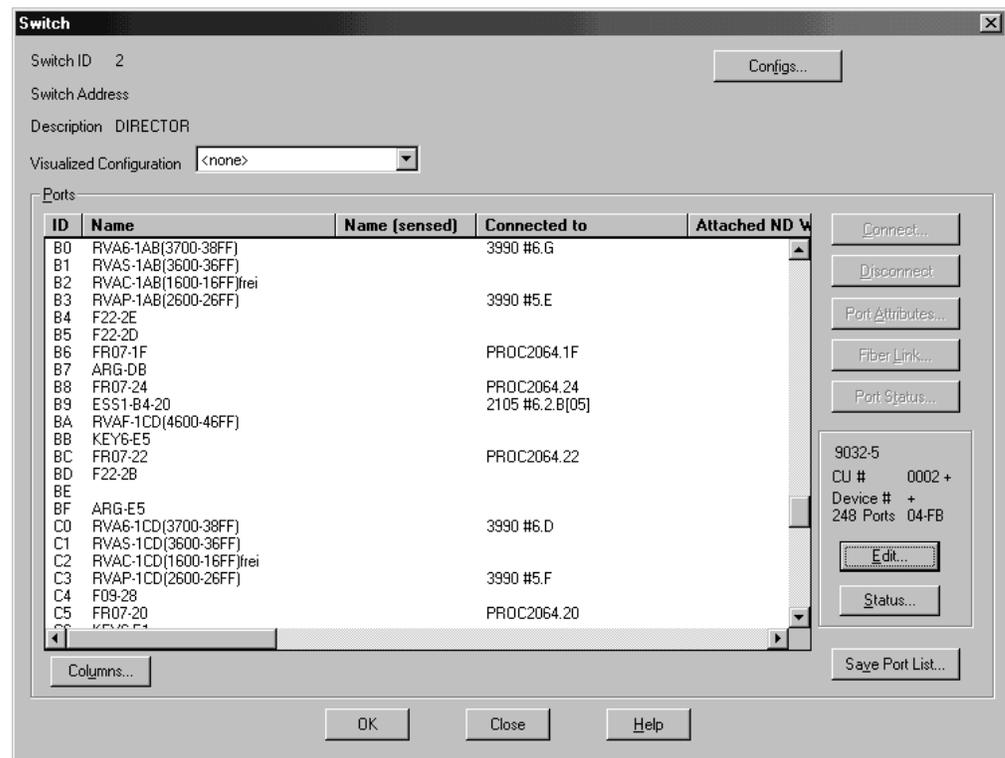
Figure 103. Switch Dialog

From the **Switch** dialog shown in Figure 103, select one or more unconnected switch port(s) from the Ports list and press the **Connect...** button. If you select multiple ports, the **Select Multiple Connection Target(s)** dialog appears, allowing you to select the connection target interfaces.
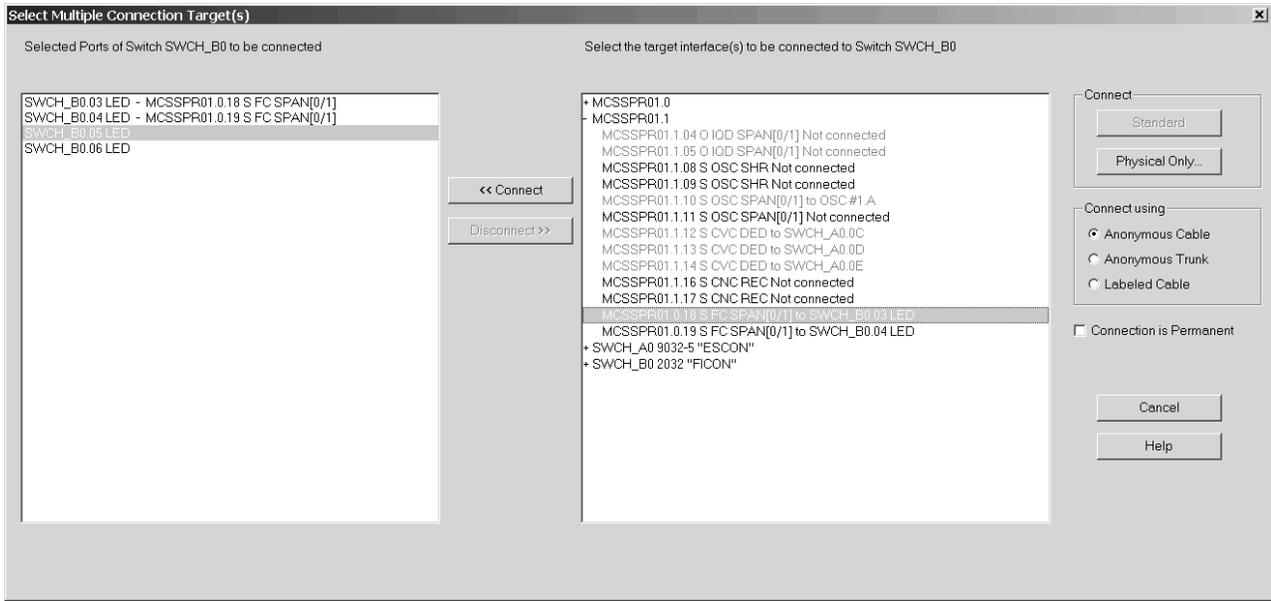
Figure 104. Select Multiple Connection Target(s)

In the right list box, you can double-click on a target object with a plus (+) sign to navigate to its interfaces. Targets that are not eligible for connection are disabled.

In the left list box containing the previously selected switch ports, select one of them and then select an eligible connection target from the right list box (a processor CHPID or another switch port). These selections enable the *<<Connect* button. Pressing this button copies the selected connection target from the right list box into the left list box, thus displaying a pair of connection origin and target.

Repeat this process for every connection that you want to establish. After you have specified all necessary connections in the dialog, press the **Physical Only** button.

Pressing the **Physical Only** button causes HCM to attempt to make the physical connection between the selected switch port and target. If you are connecting switch ports to CHPIDs, and if the switch ID of the selected switch ports does not coincide with the dynamic switch defined with the CHPID, then HCM shows the *CHPID's Dynamic Switch Modification* dialog (Figure 105 on page 175) where you can update the dynamic switch for all involved CHPIDs.
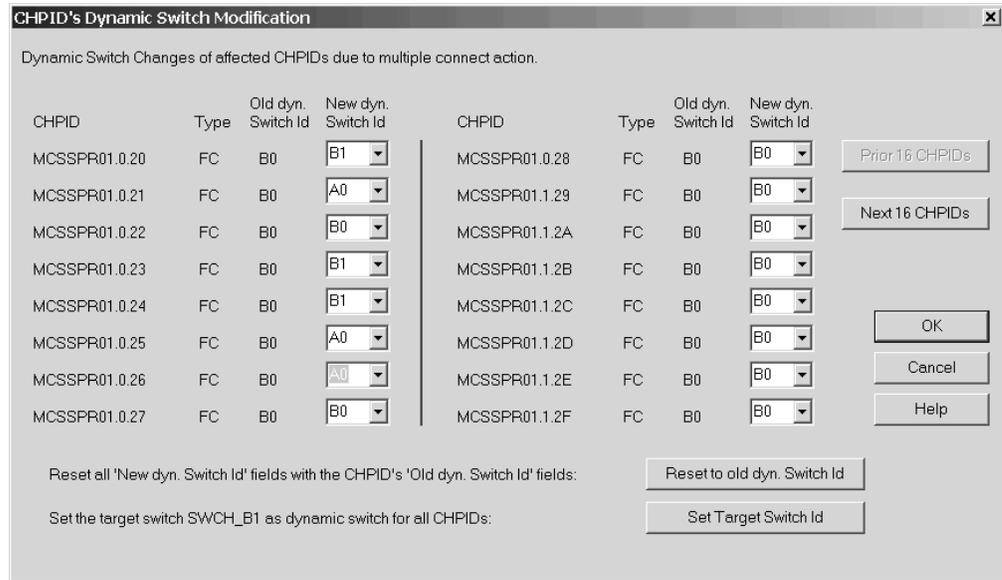
Figure 105. CHPID's Dynamic Switch Modification Dialog

Disconnecting the switch port from a target interface

Open the **Switch** dialog by double-clicking on a switch in the configuration diagram, or by choosing **Switch...** from the **Edit** menu. Select the connected switch port from the Ports list box and press the **Disconnect** button. HCM will ask you to confirm the disconnection before doing it.

Connecting CHPIDs via CTC connections

You can create a CTC (channel to channel) connection between two CHPIDs on the same or different processors, either directly or through a switch. When connecting through a switch, both CHPIDs must be able to access the same dynamic switch. One CHPID must be CNC or FCV, the other CTC in case of an SCTC connection. For FCTC connections, two FC CHPIDs have to be connected.

If you want to have a CTC connection involving spanned CHPIDs, you must establish a CTC connection between each CHPID instance that should be involved in the CTC connection.

HCM considers a CTC connection as a single entity comprising selected control units and I/O devices connected to the CHPIDs. The following diagram represents two CTC connections, connecting two CHPIDs through a switch. Note that the CTC controllers and devices are not explicitly shown.
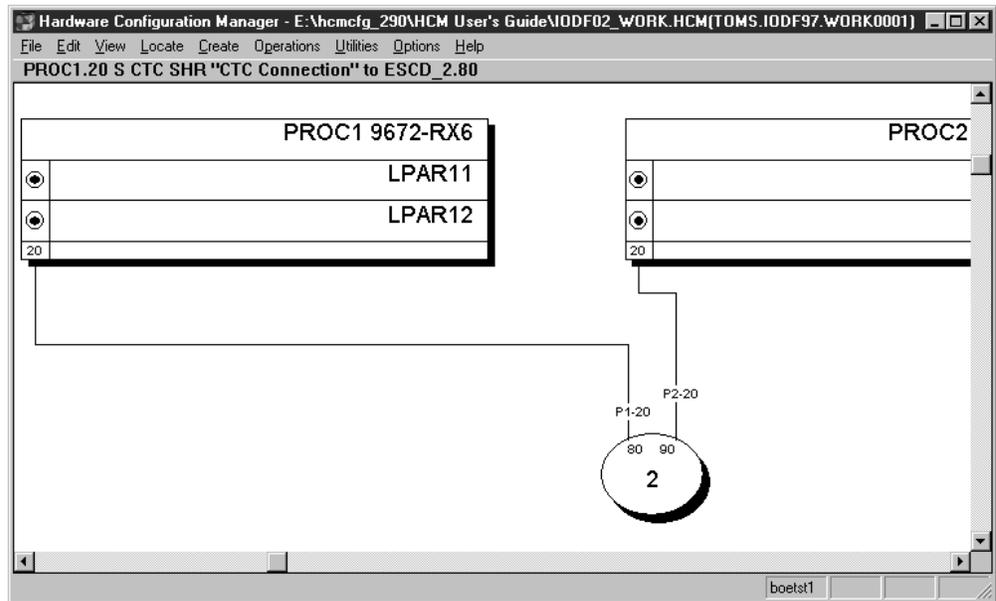
Figure 106. CTC Connection Between Two CHPIDs on Different Processors Through a Switch

Creating a CTC connection

Choose **CTC Connection...** from the **Create** menu or press **Create...** in the **CTC Connections** dialog to open the **Create CTC Connection** dialog. Select the two CHPIDs at either end of the CTC connection by pressing the appropriate **CHPID...**button and choosing a CHPID from the resulting list box. Double-click on a processor in the list with a plus (+) sign to expand it to display all possible target interfaces (CHPIDs); double-click on a processor with a minus (–) sign to collapse it. Select a CHPID and press **OK**; alternatively, double-click on the CHPID.
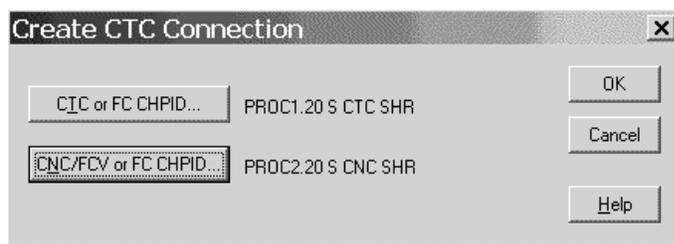
Figure 107. Create CTC Connection Dialog

Selecting **OK** will validate whether or not the CHPIDs may be involved in a CTC connection. One of the following conditions must be met:

- Neither of the CHPIDs are physically connected to anything, (for a new point to point), or
- Both of the CHPIDs are connected to a switch, both CHPIDs have the same dynamic switch defined, and both CHPIDs have physical access to the dynamic switch either directly or via at least one port-to-port connection from the entry switch to the dynamic switch. If a CHPID is on a chained switch, and there is more than one possible route between the chained switches (that is more than one choice for the exit link address), the user will be prompted to choose one, or

- (usual case) The CHPIDs are already physically connected to each other, but neither CHPID has any control units defined, (to fix up a point-to-point).

If none of the above conditions are met, you will be informed and the dialog will remain displayed so that different CHPIDs may be chosen.

Once this has been done, the **Create CTC Connection** dialog will disappear, and be replaced with either the **Accept Proposed CTC Control Units** dialog or the **Edit CTC Connection** dialog, allowing the user to specify the control units and devices used in the connection.

Editing a CTC connection

Since CTC connections are not displayed as visible objects in the configuration diagram (only their cables are shown), they cannot be selected directly for editing. Choose **CTC Connections...** from the **Edit** menu to open the **CTC Connections** dialog. Select a connection from the list and press **Edit...** the **Edit CTC Connection** dialog appears. Alternatively, after you have created a CTC connection, pressing **OK** in the **Create CTC Connection** dialog automatically displays the edit dialog.
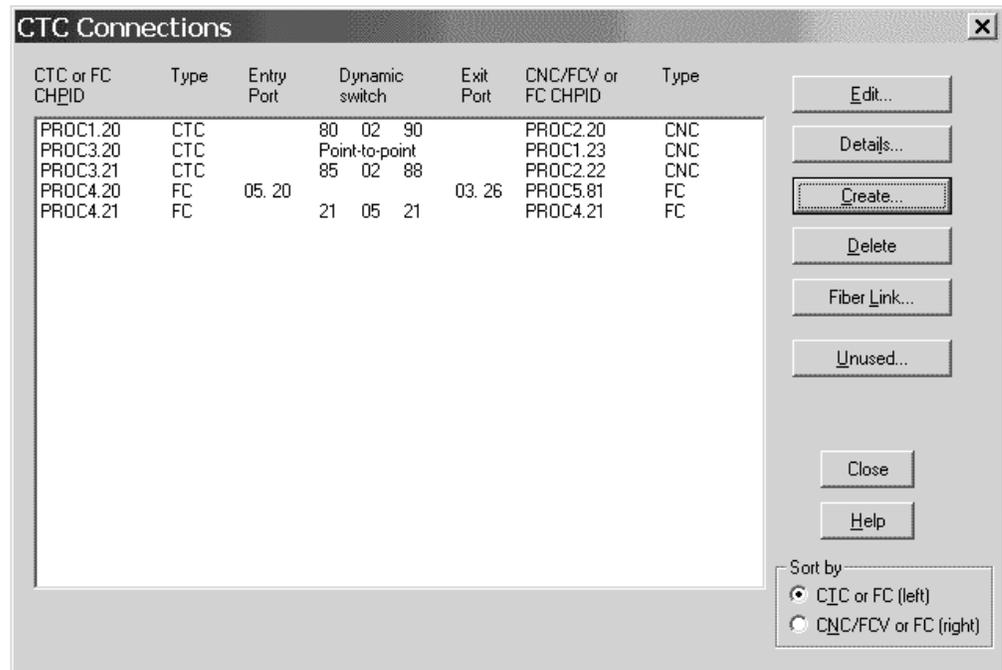
Figure 108. CTC Connections Dialog

In the **Edit CTC Connection** dialog, you can view all control units and I/O devices comprising the selected CTC connection. The dialog shows two lists, side by side.

Edit CTC Connection

CTC connection routed via SWCH_2

Control units on PROC1.20 S CTC SHR

Control Unit	Devices	UA OS	Destination on PROC2
4110	4110,4	00	LPAR11
4120	4120,4	00	LPAR12

Edit... | CU<>Proc... | I/O Devices...
Add New... | Add Existing... | Remove

Control units on PROC2.20 S CNC SHR

Control Unit	Devices	UA OS	Destination on PROC1
5110	5110,4	00	LPAR11
5120	5120,4	00	LPAR12

Edit... | CU<>Proc... | I/O Devices...
Add New... | Add Existing... | Remove

Cancel | Help

Figure 109. Edit CTC Connection Dialog

The left-hand listbox shows all of the control units connected to the CTC CHPID, or FC CHPID in case of an FCTC connection, that represent images available via the CNC or FCV CHPID, or FC CHPID in case of an FCTC connection.

Note: For purposes of this discussion, the term "image" refers to either a partition on an LPAR host, or the processor itself if it is running in BASIC mode.

There is at least one entry per image, even if there is no control unit connected that represents the partition, in which case all data columns except "Destination on ...." will be blank. If the control unit has a discontinuous range of devices connected to it, or if the device range is made up of a mixture of CTC and BCTC devices, then there will be one or more subentries under the main entry that show additional device ranges. For instance, if control unit 4110 had devices 4110-4113 and 4116-4117 connected (discontinuous), and control unit 4120 had devices 4120,4 (CTC) and 4124,4 (BCTC), the entries would look like this:

4110	4110,4	00	LPAR11
	4116,2	06	
4120	4120,4	00	LPAR12
	4124,4 (B)	04	

Each control unit or image comprises a single selectable row, even if it consists of multiple lines of text showing different device ranges; for instance, selecting either the 4110,4 line or the 4116,2 line will select the entire 2-line entry for CU 4110, which represents image PART11.

The right-hand listbox is identical; the only difference is that it displays all of the control units connected to the CNC or FCV CHPID, or FC CHPID in case of an FCTC connection, that represent images available via the CTC CHPID, or FC CHPID in case of an FCTC connection.

For images that have no control units to represent them, you may click **Add New...** to create new control units and devices to represent the image, or **Add Existing...**

to select an existing CTC control unit that may be connected to the CHPID so as to represent the selected image. For images that have control units, you may click **Edit...** to open the **Edit Control Unit** dialog, **CU <=> Proc...** to open the **Control Unit <=> Processor Definitions** dialog, or **I/O Devices...** to open the I/O Device List dialog. You may also choose to disconnect the control unit from the CHPID (and possibly delete the control unit and its devices) by clicking **Remove**.

Note: The **Edit CTC Connection** dialog can sometimes be preceded by the **Accept Proposed CTC Control Units** dialog. It will appear in those cases where HCM finds that not all images in the connection are represented by control units, and where there are CTC control units already in the IODF that would be suitable to represent those images based on current usage. These control units will appear in gray in the listboxes. After you accept or reject these proposed CHPID-control unit connections, the **Accept Proposed CTC Control Units** dialog will be replaced by the normal **Edit CTC Connection** dialog.

Viewing a connectivity report for a CTC connection

Select the connection from the **CTC Connections** and press **Details...**, the **View CTC Connection Details** dialog will appear, showing all image-to-image connections showing the control units and devices that each image uses to access the other image.
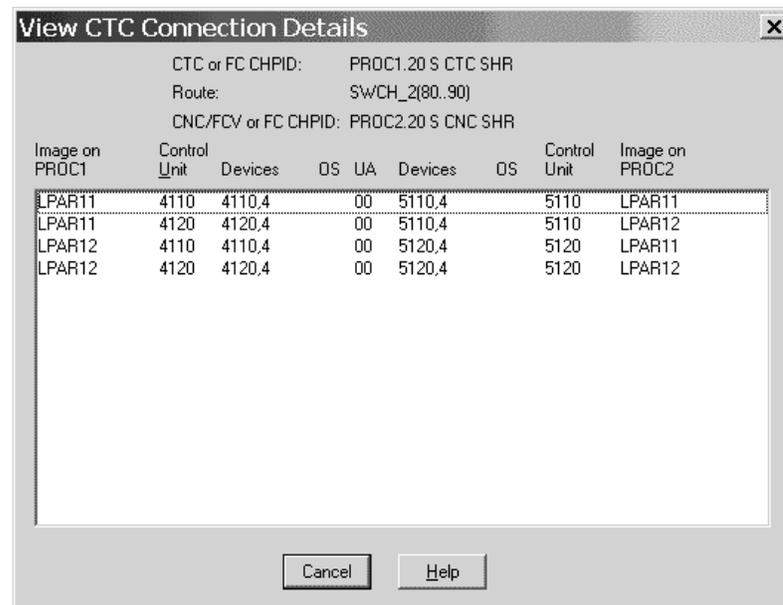
Figure 110. View CTC Connection Details Dialog

It is possible for each image-to-image connection to span multiple listbox entries. As an example, if all of the images used in the example above used 4 CTC devices and 4 BCTC devices, there would be 12 listbox entries, the first 4 showing:

LPAR11	4110	4110,4	00	5110,4	5110	LPAR12
LPAR11	4110	4114,4 (B)	04	5114,4 (B)	5110	LPAR12
LPAR12	4120	4120,4	00	5120,4	5120	LPAR11
LPAR12	4120	4124,4 (B)	04	5124,4 (B)	5120	LPAR11

Furthermore, this listbox will have entries to show devices that use unit addresses not mirrored on the destination image. These entries will show blanks in control

unit and device columns on either the left or right side. To illustrate, assume LPAR12 is excluded from device 5112's explicit candidate list. The first listbox item would be split into three entries as follows:

LPAR11	4110	4110,2	00	5110,2	5110	LPAR12
LPAR11	4110	4112,1	02			LPAR12
LPAR11	4110	4113,1	03	5113,1	5110	LPAR12

This shows that LPAR11 is able to communicate to LPAR12 via unit address 02, but LPAR12 has no device defined on this UA to communicate back.

Deleting a CTC connection

Select the connection from the **CTC Connections** and press **Delete**. All of the control units that are connected to the CHPIDs for this connection will be disconnected. The control units that no longer have any CHPID connections will be deleted along with their attached devices.

Editing the fiber links for CTC connections

You can edit the fiber link for a point-to-point CTC connection. Open the **CTC Connections** dialog by choosing **CTC Connections...** from the **Edit** menu. Choose the point-to-point connection and press the **Fiber Link...** button to open the **Edit Fiber Link** dialog. You can then edit the composition of the Fiber link between the CHPIDs. For more details, see "Editing fiber links" on page 192.

Note that CTC connections through switches have two fiber links, one on each switch port. To edit those fiber links, open the **Switch** dialog by double-clicking on the switch in the configuration diagram. Select the desired switch port and click on the **Fiber Link...** button to display the **Edit Fiber Link** dialog.

Viewing and deleting unused CTC control units, devices, and connections to CHPIDs

It is possible that the IODF contains CTC control units, devices, or connections between CHPIDs and CTC control units that cannot be attributed to any valid CTC connection. This can happen in the following circumstances:

- Unconnected CTC control units or devices were created using HCD followed by an IODF load or resync.
- Connections via switches were made using HCD without specifying CHPID-port or port-CU links followed by an IODF load or resync.
- Only one side of a logical CTC connection was made using either HCD or HCM.
- An action within either HCD or HCM was made that affects the validity of existing CHPID-control unit connections, such as unlinking a port from a CHPID or control unit, deleting a partition involved in a CTC connection, or changing the mode of a CHPID involved in a CTC connection from shared to non-shared or vice-versa.

Under these circumstances, the **Unused...** button will appear on the **CTC Connections** dialog. Clicking this button will open the **View Unused CTC Control Units, Devices, and CHPID Connections** dialog.
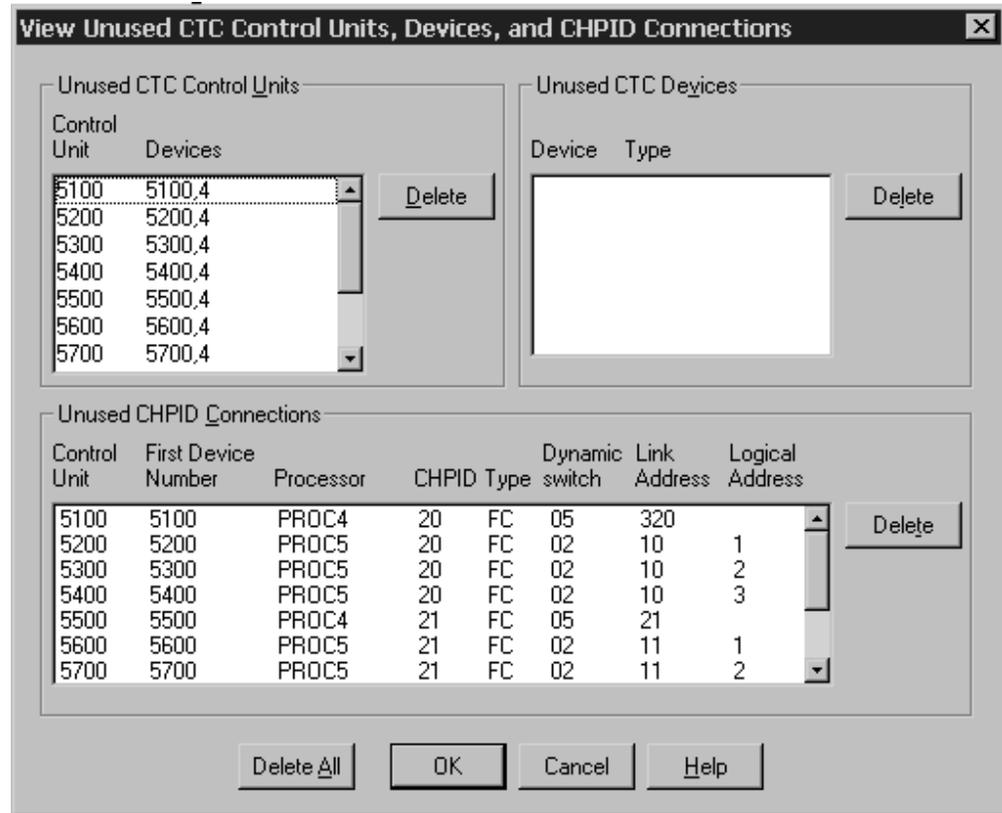
Figure 111. View Unused CTC Control Units, Devices, and CHPID Connections

All of the invalid (that is not involved in a valid CTC connection) CTC-related IODF objects will appear in this dialog. Usually the easiest way to recover from these situations is to click **Delete All** to get rid of the invalid items, and re-create the CTC connections that were affected.

To prevent this dialog from ever having to make an appearance, the following precautions should be adhered to:

- Before loading an IODF, make sure that all control units and CHPIDs involved in CTC connections via switches have appropriate links to switch ports, and that all control units involved in new point-to-point CTC connections have serial numbers specified to match up the two sides of the connection. Also ensure that unconnected CTC control units and devices are deleted. See also "Preparing the IODF" on page 72.
- Whenever you create or modify a CTC connection in HCM, make sure that both CHPIDs (that is both sides of the **Edit CTC Connection** dialog) have one control unit connected to represent an image on the other CHPID.

Validating CTC connections

Several actions can influence your CTC connections. Even when you have correctly set up a CTC connection using HCM, actions in your configuration that you perform after defining your CTC connection might influence your CTC connection in a way so that the CTC connection does no longer work. Among these actions are the following:

- Updating the access/candidate lists for a CHPID
- Changing a CHPID's mode (SHR/DED)

- Changing an LPAR number
- Deleting an LPAR
- Adding an LPAR
- Disconnecting/reconnecting a CHPID
- Deleting a CHPID
- Deleting a switch.

Whenever you make one of these actions, you should verify your CTC connections in HCM. For instance, when you create a CTC connection via HCM, the CUADD values representing logical partitions are automatically correctly set by HCM/HCD at the time the CTC connection is created. If you now modify the LPAR number, the CUADD value stored does no longer represent the chosen logical partition, as the LPAR number and CUADD no longer coincide. Though correctly defined before, the CTC connection will no longer work because of an action that influences the CTC connection.

CTC connection report

HCM offers you the possibility to create a CTC connection report via the **Print Reports** dialog, reachable via the **File** menu. You can see an example of such a report in "Sample reports and context diagrams" on page 230 (Channel-to-Channel Connections). The CTC connection report contains information about all CTC connections known to HCM in your configuration. At least before building a production IODF, use this report to carefully verify that all the CTC connections that you have defined before, are still existing as previously defined.

View CTC connection details

Besides the HCM CTC connection report, you can also verify the CTC connections defined in the configuration via the **CTC Connection Details** dialog, see Figure 110 on page 179. With this dialog, you can view the connectivity details for a selected CTC connection. This dialog shows you information about the involved processor(s) and channel paths, gives details about the route itself, and provides insight in the used images on the processor, the used CTC control units and CTC devices. In general, if blanks appear in any column in the list box (with the exception of the OS columns), then there is some gap in the connectivity and the two images will not be able to communicate using the unit address shown. Even though CTC definitions have been correctly defined before, you might find gaps indicating that some mismatch occurs due to one of the above mentioned actions.

View unused CTC control units, devices, and CHPID connections

Use this dialog to view (and clean up, if necessary) unused CTC control units and devices, or connections from CHPIDs to CTC control units that are not used in any CTC connection. To display this dialog, press the **Unused...** button on the **CTC Connections** dialog. Note that this button is available only if there are unused items to examine. When using HCM to define CTC connections, this dialog will only be available if CTC connections are incomplete. Therefore, the appearance of the **Unused...** button on the **CTC Connections** dialog serves as an indication that something is amiss with the logical definitions that make up the configuration's CTC connections. See also "Viewing and deleting unused CTC control units, devices, and connections to CHPIDs" on page 180.

Create / establish CTC connections after loading an IODF

After loading an IODF, HCM is able to detect CTC connections. In general, HCM recognizes CTC connections if all control units composing a CTC connection through a switch are connected to switch ports. Point-to-point connections are recognized if the involved control units have the same serial number.

In case the involved control units of a point-to-point CTC connection do have different serial numbers, HCM can not detect that CTC connection. In order to establish such a CTC connection so that also HCM recognizes these CTC connections, you can do the following steps. Note that it is assumed here that correct point-to-point connections have been defined in the IODF via HCD.

- Create the CTC connection under focus via the **CTC Connections** dialog where you press the **Create...** button in order to obtain the **Create CTC Connection** dialog.
- In the **Create CTC Connection** dialog, select the CHPIDs that are involved in the CTC connection, and press the **OK** button.

Now, the **Accept Proposed CTC Control Units** dialog appears showing the CTC control units to which the selected channel paths are already connected. Either accept the proposal via the **Accept** button, or reject it with the **Reject** button. In case you have accepted the proposal, the **Edit CTC Connection** dialog appears, prefilled with the accepted proposed control unit and device information. Check the information in this dialog, and if the CTC connection is already as it should be, press the **Close** button. The CTC connection is now defined, and HCM also recognized this CTC connection. In case you have rejected the proposed information, you should be able to select the correct control units and devices via pressing the **Add Existing ...** button on the **Edit CTC Connection** dialog. Select the correct CTC control units and the connected CTC devices in the **Select CTC Control Unit** dialog, press the **OK** button, and return to the **Edit CTC Connection** dialog. After having selected all CTC correct control units and devices, the CTC connection is complete. Note that under the above assumption, HCM either proposes the correct CTC control units, or you find the CTC control units and devices in the **Select CTC Control Unit** dialog.

Connecting CHPIDs via coupling facility connections

A *coupling facility* is a special logical partition that provides high-speed caching, list processing, and locking functions in a processor. It involves connecting a partition on one processor to a partition on another processor.

The following diagram represents two CF-connections involving CHPIDs 70 and 10 respectively 71 and 11. Note, that the CF control units and devices are not explicitly shown.

You can determine if a selected partition will be used solely for standard operating system connections (OS), as a coupling facility (CF) or both (CF/OS) by selecting a *Usage* option in the **Create Partition** dialog from the **Create** menu.
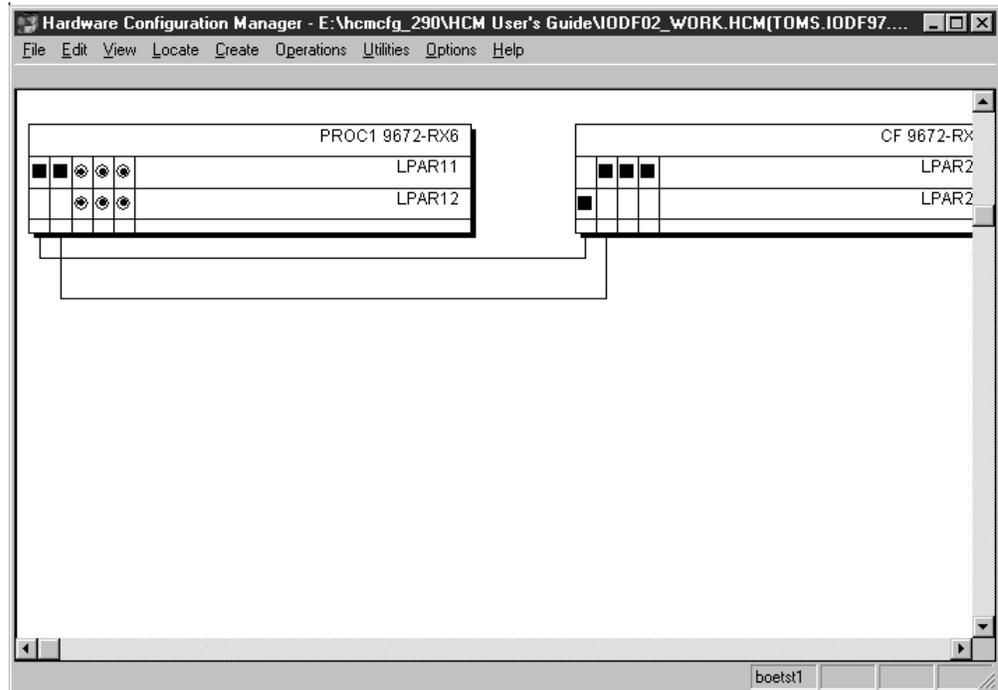
Figure 112. Coupling Facility Connection

You can only create a coupling facility connection between a sending CF CHPID on one processor or partition and a receiving CF CHPID on another. You can view all coupling facility connections in the dialog of the same name. Open the **Coupling Facility Connections** dialog with the **CF Connections...** option in the **Edit** menu, and use the View section options to sort the list box contents by CF sender or receiver CHPID.
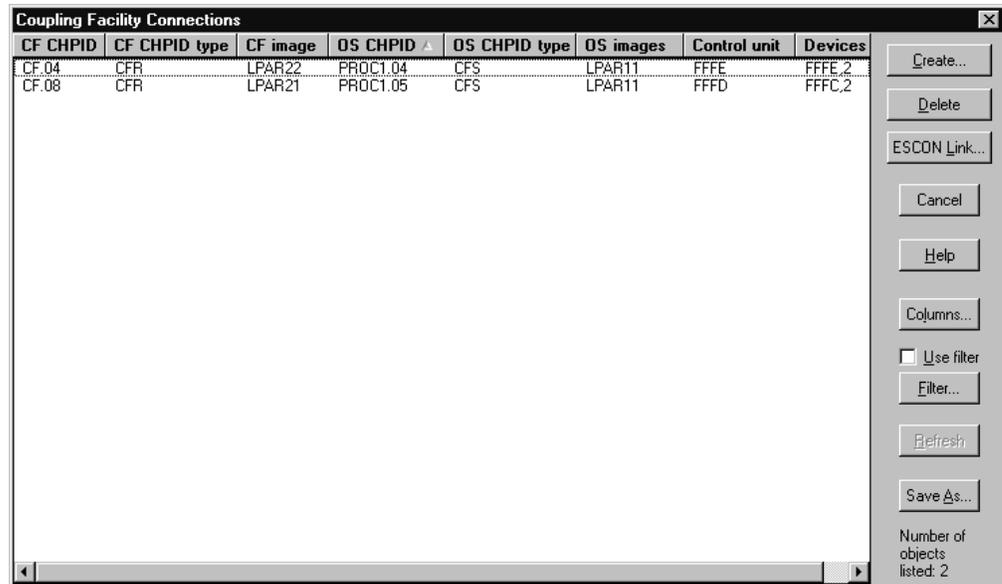
Figure 113. Coupling Facility Connections Dialog

Creating a coupling facility connection

Choose **CF Connection...** from the **Create** menu or press the **Create...** button in the **Coupling Facility Connections** dialog to open the **Create Coupling Facility Connection** dialog, which allows you to specify the sending and receiving channels for the connection.
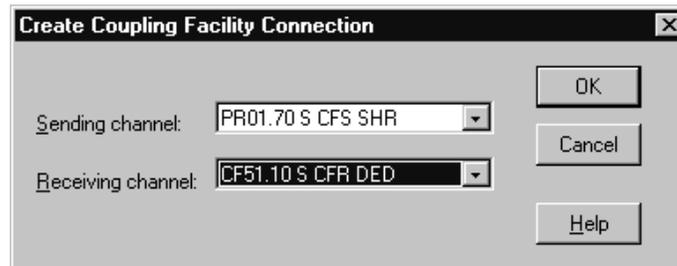
Figure 114. Create Coupling Facility Connection Dialog

Pressing **OK** will automatically open the **Choose CU and Device Numbers** dialog, where you can set the control unit number and device number for the connection. If there already is a CF connection between the two partitions, then the same control unit is used and the control unit number cannot be edited. Each CF connection requires devices of its own.
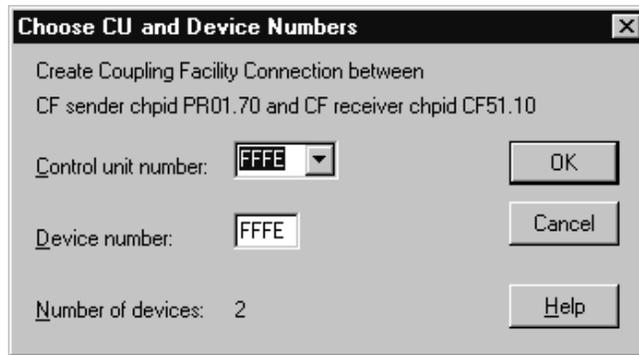
Figure 115. Choose CU and Device Numbers Dialog

Deleting a coupling facility connection

Select the coupling facility connection from the list box in the **Coupling Facility Connections** dialog and press the **Delete** button. HCM will display a confirmation dialog before "deleting" the connection; that is, HCM disconnects the two CHPIDs and deletes the corresponding control units and devices.

Chapter 8. Physical connections

This chapter explains how to add strictly physical elements to the connections of your configuration; specifically: cables, patchports, and crossbar switches.

In HCM, connections are made initially without concern for physical cables and patchports (see Chapter 7, "Connecting objects in your configuration," on page 155). Cables and patchports are assigned to connections after the connections have been made. These physical aspects of the connection are not reflected in the IODF and are not required in order to build a correct production IODF.

Managing physical cables

New connections initially have no physical cables assigned. You can either assign a cable immediately after creating the connection, or assign cables after a series of connections.

HCM displays cable IDs and length information in the Object Information Bar (at the top of the screen) when the mouse cursor is placed over an interface with an assigned physical cable. This information will appear in reports and on printed configuration diagrams.

HCM interprets trunk cable IDs as two fields separated by a slash, '/'. The first field identifies the trunk; the second indicates the pair of conductors (channel) within the trunk. For example, trunk cable ID TR1234/07 is Channel 07 in Trunk TR1234.

Cables can have three states: *in use, unused or in inventory*.
* **In Use:** Cables that are currently assigned to connections. Cables are assigned to connections using the **Assign Cable** dialog.
* **Unused:** Cables that were once assigned to connections. A cable becomes unused when its connection is broken or when a new cable is assigned to the connection. Unused cables retain information about where they were last used, to make it easy to identify them when you re-connect them. Note that re-connecting an interface or re-positioning it on a daisy-chain will disconnect the physical cable on the interface.
* **In Inventory:** Cables that are not currently assigned to a connection and for which there is no record of any previous connection they might have been assigned to. They can be assigned to connections.

Cables come to be in inventory in two ways.
* Unused cables can be moved to inventory using the **Cable Assignments** dialog. The only difference between unused cables and cables in inventory is that unused cables retain information about their last connection. You would move unused cables into inventory when you know that they will not be re-used in the near future.
* The inventory can be populated by importing cable descriptions from an ASCII file. See Chapter 15, "Importing and exporting data," on page 317 for more information.

An unused cable retains its last connection information until it is again assigned to a connection, or it is deleted, or it is placed into inventory. A cable in inventory, remains there until it is assigned to a connection or it is deleted.

Assigning cables to connections

There are several ways to edit the cable assignments for connections in your configuration.

The most efficient way to edit the cable assignments for several connections at once is to select **Cable Assignments...** from the **Edit** menu. This opens the **Cable Assignments** dialog which lists the connections and their cable assignments.
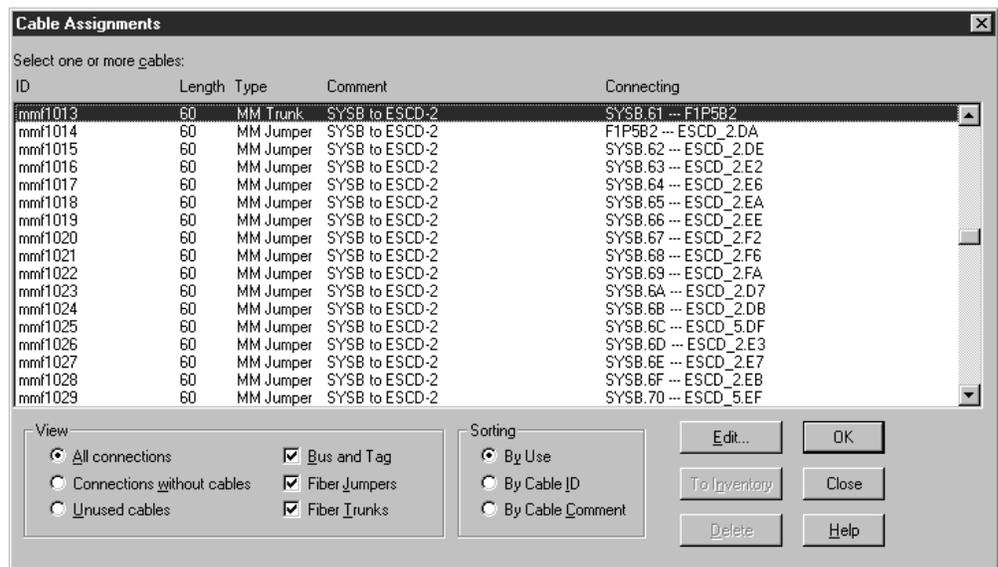
Figure 116. Cable Assignments Dialog

Viewing cables and connections

In the **Cable Assignments** dialog, use the View options to determine which cables and/or connections appear in the list box. You can display all connections, regardless of whether physical cables are assigned to them. Connections are ordered by CHPID. You can display connections without cables, or all unused cables and cables in inventory.

The Bus and Tag, Fiber Jumpers and Fiber Trunks options control which types of cables are displayed in the list box. The contents of the list box are unaffected by these options if the "Connections without cables" option is selected.

You can also specify the order in which cables and connections are listed with the Sorting options.
- Choose "By Use" to sort the connections by the names of the connected interfaces. Cables that form a daisy- chain are grouped together with the second and subsequent cables in the chain indented.
- Choose "By Cable ID" to sort connections by cable ID. Connections without cables appear first in no particular order, followed by connections with cables in alphanumeric order.

Creating or editing cables

Select either "All connections" or "Connections without cables" in the View section of the **Cable Assignments** dialog, then select the connection(s) and press the **Edit...** button. The **Assign Cable** dialog is opened allowing you to create or edit the cable assigned to the connection.

Transferring unused cables to inventory

Select "Unused cables" in the View section of the **Cable Assignments** dialog, then select the cable(s) to be moved to inventory. Press the **To Inventory** button. Note that cables in inventory, as well as unused cables, appear as possible choices when editing cable assignments in the **Assign Cable** dialog.

Deleting cables

Select the "Unused cables" option in the View section of the **Cable Assignments** dialog, then select the cable(s) and press the **Delete** button. HCM displays a confirmation dialog before deleting the selected unused or inventory cables. HCM does not retain any information on these cables, once they are deleted.

Other ways to assign cables

Besides using the **Cable Assignments** dialog, you can use any of the following methods of accessing the **Assign Cable** dialog.

- To edit a single cable assignment from the diagram, select the lower interface of the connection, and select **Edit cable on selected interface...** from the **Edit** menu (Shortcut: press F5).

- To edit several cable assignments from the diagram, choose **Interface Selection...** from the **Options** menu and check "Edit the physical cable assignment for the interface". Now double-click on the lower interface of each connection to open the **Assign Cable** dialog.

- You can edit the cable assignments for connections within a fiber link from the **Edit Fiber Link** dialog by selecting a cable or dashes (unlabeled cable) and clicking on the **Edit Cable...** button. See "Editing fiber links" on page 192 for more details.

Creating and editing cables

Use any of the methods described under "Assigning cables to connections" on page 188 to open the **Assign Cable** dialog for a particular connection.
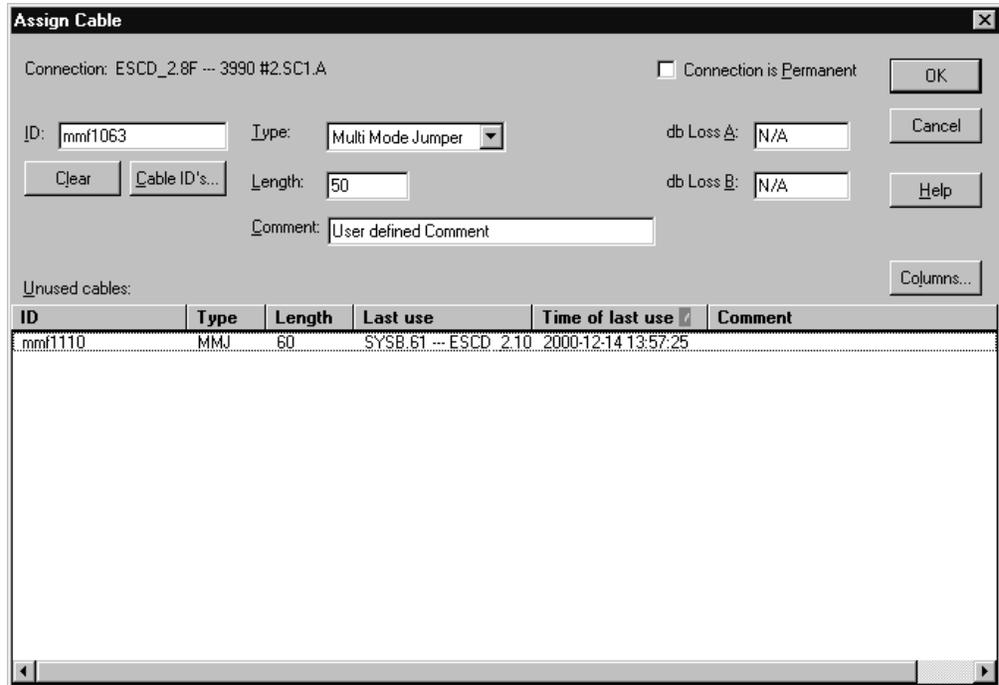
Figure 117. Assign Cable Dialog

Creating a new cable to assign to a connection

In the **Assign Cable** dialog, you can create a new cable for a connection by entering a new Cable ID, cable type, and (optionally) the cable's length and db loss, then pressing **OK**. The ID must be unique.

HCM currently supports the following cable types:

Parallel

> Gray Fully Populated
> Gray Partially Populated
> Blue Fully Populated
> Blue Partially Populated
> Bi-directional

Serial

> Single Mode Jumper
> Multi Mode Jumper
> Single Mode Trunk
> Multi Mode Trunk

Reusing an existing cable for a connection

The Unused Cables listbox contains both unused cables and cables in inventory. The unused cables are listed first. To reuse an existing cable for a connection, select it by clicking on it in the Unused Cables list box in the **Assign Cable** dialog. (Alternatively, type its cable ID in the appropriate text box, and press **OK**).

Pressing the **Cable ID's...** button displays a list of all cable IDs already in use by connections.

When you change the cable assignment for a connection, the old cable is placed in the unused cables pool.

Removing a cable from a connection

To remove the current cable assignment from the connection, select "No physical cable assignment" in the listbox and press **OK**. The cable goes into inventory and the connection is left with no cable assignment.

Printing cable labels

You can print labels for cables that are in use in your configuration. Cable labels can then be attached to the physical cables for easy identification. Open the **Select Cable Labels to Print** dialog by choosing **Print Cable Labels...** from the **File** menu.
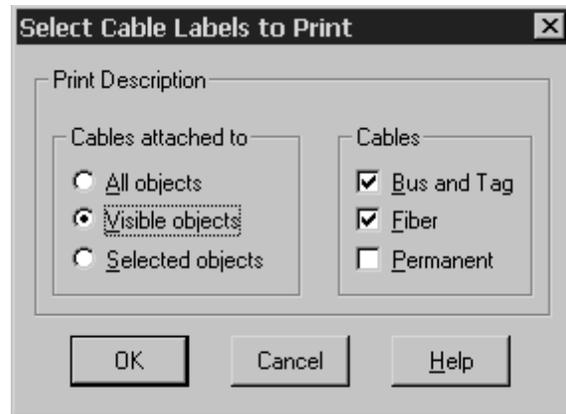
Figure 118. Select Cable Labels to Print Dialog

Select the type of cables (bus and tag; fiber; permanent connections to patchports) for which you wish to print labels. You can restrict the number of labels by printing only those attached to visible objects (those currently included in the diagram), or only those that are currently selected.
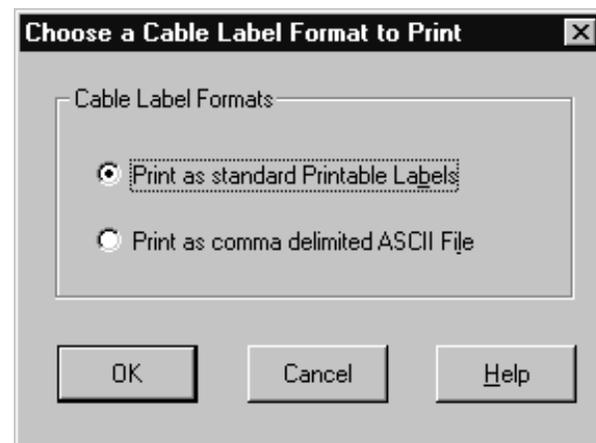
Figure 119. Choose a Cable Label Format to Print Dialog

Pressing **OK** opens the **Choose a Cable Label Format to Print** dialog. You can print labels directly to the printer with the standard format, or to an ASCII file where you can customize the formatting.

- **Standard printable labels** prints the labels to the current printer, which is presumably loaded with appropriate label forms. These self laminating labels must comply to the following specifications:

Color	Clear
Write-on area (width x length)	
	1.00 x 0.47
Total label (width x length)	1.00 x 1.33
Number of columns	7
Number of rows	7
Labels per sheet	49

- **Comma delimited ASCII file** prompts you for the filename to write the information to. There are programs available that can read, format and print the cable label information.

See "Sample cable labels" on page 265 for examples of standard printable labels and comma delimited ASCII file labels.

Editing fiber links

As described in Chapter 7, "Connecting objects in your configuration," on page 155, a logical connection may be "realized" by more than one physical cable, often passing through a number of patchports. For serial connections, HCM uses the term "fiber link" to describe the complete connection between two interfaces (optical transceivers). HCM allows you to insert patchports into an fiber link, to assign cables to each segment of the link, and to mark certain connections as "permanent" so they will not disappear when connections between active ESCON or FICON objects are broken. There are two methods for setting up fiber links. For "infrastructure" connections (i.e., trunk cables), you should use the **Cabinet** dialog. To document the jumper cables and patchports that are used in a connection between active ESCON or FICON objects, you should use the **Edit Fiber Link** dialog. In general, you should use the following sequence:

1. Create any cabinets and patchports and make necessary permanent connections between them. This is normally done when new cabinets are installed and connected with trunk cables.
2. Make permanent connections from patch panels to CHPIDs and switch ports.
3. Make all logical connections (between CHPIDs, switch ports, and channel interfaces). This step can be performed separately and without regard for fiber links.
4. Add patchports and cables to the fiber link (that is, make mutable connections by editing the fiber link).

This section covers steps 1 and 2 (Infrastructure Connections) and 4 (Jumper Connections), plus information on disconnecting fiber links. For more information on step 3, see Chapter 7, "Connecting objects in your configuration," on page 155.

Infrastructure connections

HCM has the concept of a "permanent" connection to model infrastructure connections that are set up as a permanent fixture of the data center. Any serial connection that does not involve a converter may be marked as "permanent" in HCM, but normally this attribute is used for trunk connections that

- link the backs of two patch panels,
- link a group of CHPIDs to the back of a patch panel, or
- link a group of switch ports to the back of a patch panel

When a patchport is connected to a CHPID or switch port in this manner, the patchport (and any other patchports connected to it) are automatically added to the fiber link when the CHPID or switch port is logically connected. Futhermore, these connections will remain in place when logical connections are broken. Such connections should be made from the **Cabinet** dialog. As an example, to connect the back of one panel to the back of another, start by double-clicking the cabinet in the configuration diagram (or choose **Cabinet...** from the **Edit** menu and double-click the appropriate cabinet to open the **Cabinet** dialog shown in the following figure:

Make sure the **Back** radio button in the **View** group box is selected. This tells
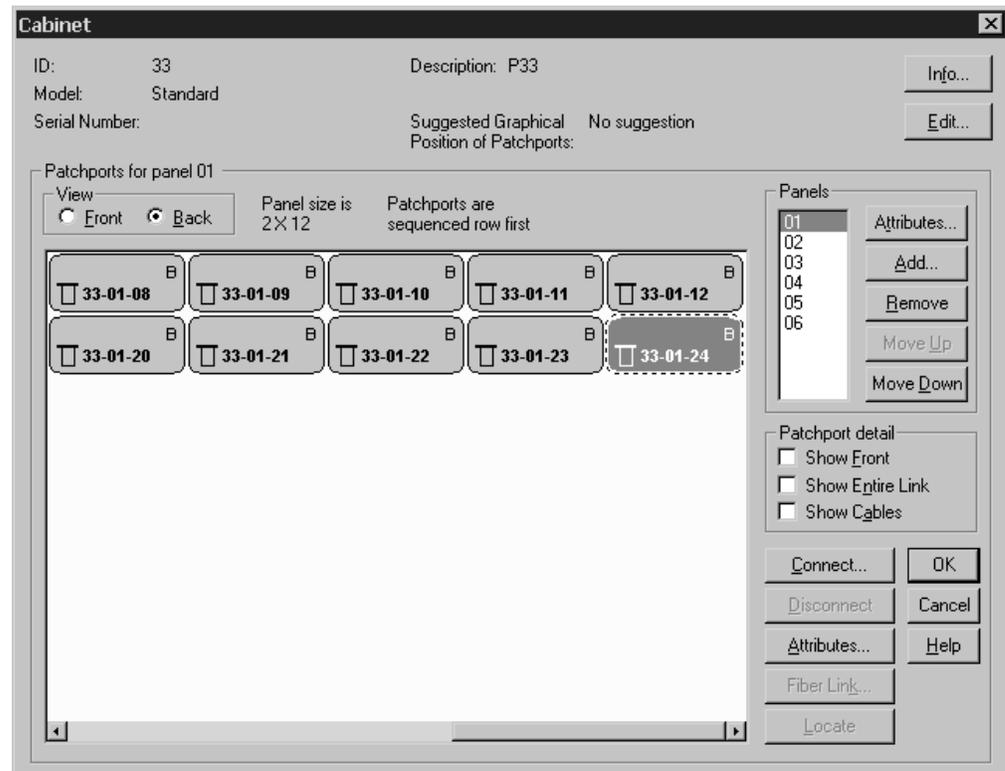
Figure 120. Cabinet Dialog

HCM that subsequent connect operations are made to the back of patchports. Select the appropriate panel from the **Panels** list, and then select all of the patchports in the panel by clicking (and holding) the left mouse button on the first patchport and moving the pointer to the last patchport before releasing. Now click **Connect...**; you will be presented with the **Select Connection Target(s)** dialog.
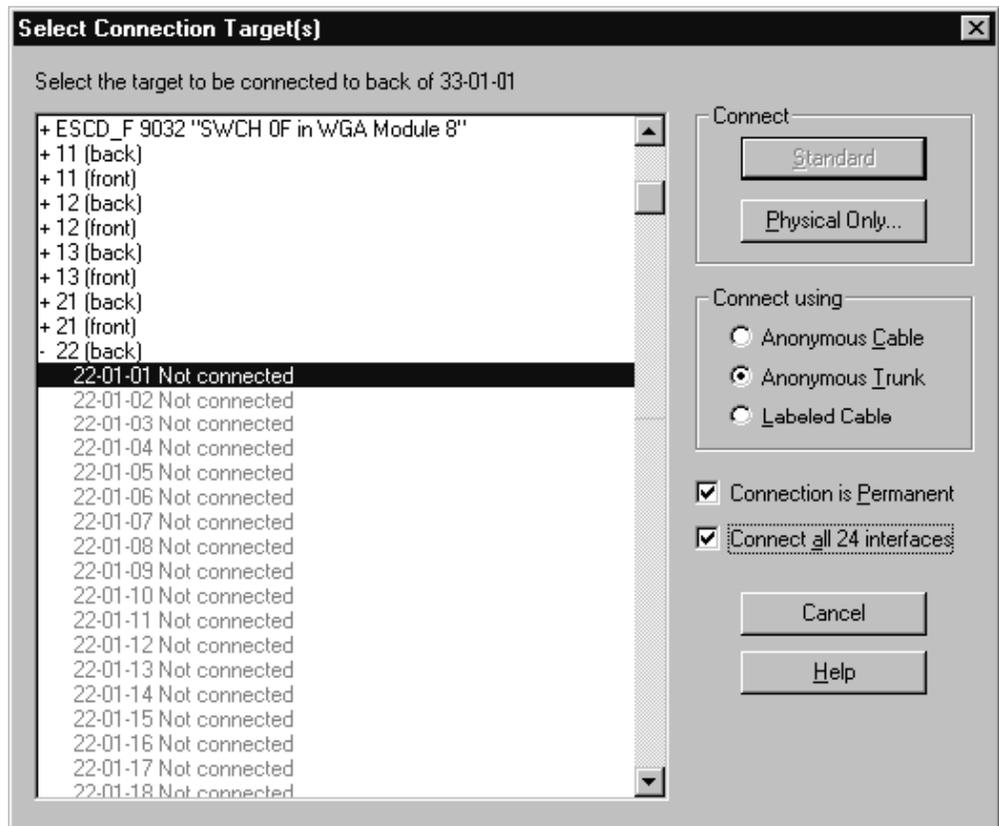
Figure 121. Select Connection Target(s) Dialog

In this dialog, check both the **Connection is Permanent** box and the **Connect all xx interfaces** box.

Note: The **Connect all xx interfaces** box is available only when connecting patchports. You can only connect all patchports in one step, if they are contiguous. Otherwise you receive an error message.

Double-click the "(back)" entry of the target cabinet, and then double-click the first patchport in the target panel. The **Cabinet** dialog reappears with the patchport diagram updated to include the permanent connections. If you want to label the cables involved in these connections, you can do so by choosing the **Labeled Cable** option. In this case, the **Assign Cable** dialog will appear as many times as necessary to allow you to label the cable for each connection. Connecting the back of a patch panel to a range of CHPIDs or switch ports is similar. In these cases, you pick the appropriate starting CHPID or port from the **Select Connection Target(s)** dialog. If any of the target interfaces are used in complete fiber links, then the patchport will be inserted into the fiber link (you will be asked to confirm the connection in such cases). This allows trunk connections to CHPIDs and switch ports to be added after an IODF load without disturbing connections in the IODF.

Disconnecting permanent connections from patchports

To disconnect one or more permanent connections on patchports: Double-click on a cabinet in the configuration diagram (or choose **Cabinet...** from the **Edit** menu and double-click the appropriate cabinet) to open the **Cabinet** dialog. Select the correct side of the panel (back or front), select the range of patchports to be disconnected, and click the **Disconnect** button on the **Cabinet** dialog panel. Since the connections

are marked as permanent, you will be asked to confirm that you want to disconnect. If any of the patchports are involved in complete fiber links, they will be removed from the fiber link when the permanent connection is broken.

Jumper connections

Jumper connections are usually made using the *Edit Fiber Link* dialog after logical connections have been set up. As an example, when you are configuring a new controller, typically you will create the controller and connect it to CHPIDs and/or switch ports as described in previous chapters. Once this is done, you can document the physical cables and patchports that are used to realize the logical connections by selecting the "bottom" end of each connection (e.g. the controller interface connected to a switch port, or a switch port connected to a CHPID) and clicking "Fiber Link..."

Editing fiber links in the dialog

Open the **Edit Fiber Link** dialog by pressing the **Fiber Link...** button in the edit dialog for the active ESCON or FICON object "below" the fiber link (this includes Edit Converter, Controller, Director, and Edit CTC Connection dialogs). See panel in Figure 122 on page 196. Note that if the **Fiber Link...** button is disabled, there is no fiber link for the selected interface; you must make a connection before you can edit the fiber link.

Alternatively, in the Cabinet dialog panel, select the patchport whose fiber link is to be edited. Open the **Edit Fiber Link** dialog by pressing the **Fiber Link...** button in the **Cabinet** dialog. Note that if the **Fiber Link...** button is disabled, there is no fiber link for the selected interface; you must make a connection before you can edit the fiber link.

The Current Fiber Link composition list box shown in Figure 122 on page 196 displays all objects in the fiber link, including cables which are indented, and undefined cables which are represented by dashes (- - -). The Available Patchports list box on the left displays all cabinets and their patchports. This box is empty unless dashes are selected in the Current Fiber link composition list box. Double-click on a cabinet with a plus (+) sign to expand it to display its available patchports; double-click a cabinet with a minus (-) sign to contract it. Invalid patchports are disabled.
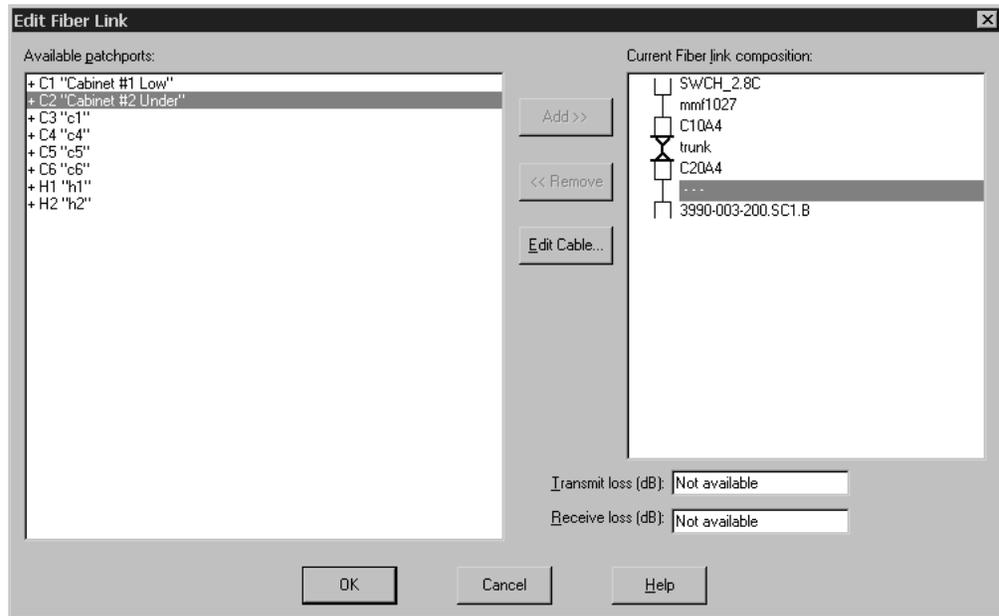
Figure 122. Edit Fiber Link Dialog

Adding a patchport to the fiber link

To add a patchport to the fiber link, select the dashes (- - -) at the position where you wish to add the patchport. All available cabinets and patchports will appear in the list box on the left; any permanently connected patchports are shown on the same line with a cable ID if there is one. Select an available patchport and click on the **Add** button. Alternatively, double-click on the patchport. If the patchport is permanently connected to others, all will be added to the fiber link. The selected patchport will be connected to the object listed below the selected dashes.

Removing a patchport from the fiber link

To remove a patchport from the fiber link, select the patchport in the composition list box and click on the **Remove** button. Alternatively, double-click on the patchport. If the patchport is permanently connected to other patchports, all are removed from the fiber link at once. Note that you cannot remove a patchport that is permanently connected to a CHPID in this dialog.

Note: You cannot remove a patchport that is permanently connected to a CHPID, switch port, or controller interface in this dialog. If you indeed intend to break such a connection, first clear the "Connection is Permanent" flag using the "Assign Cable" dialog.

Assigning cables between the objects in a fiber link

To assign cables between the objects in a fiber link, select the dashes (- - -) at the position where you wish to add the cable and press the **Edit Cable...** button. This opens the **Assign Cable** dialog. See "Managing physical cables" on page 187 for more information

You can also edit dB loss for the fiber link. Specify the transmit and receive attenuation (dB loss) for the fiber link; these fields are stored and can be compared with subsequent dB loss measurements.

Disconnecting a fiber link

When a switch port or a serial interface on a controller is disconnected (either explicitly or implicitly via reconnect), HCM will prompt the user to choose which mutable link is to be broken if there is more than one. Consider the example from Figure 123 showing four CHPIDs connected to two switches. From the diagram, it is evident that both the CHPIDs and the switch ports are trunked to the backs of cabinets, and that the link from each CHPID to its connected port also passes through another trunk connection linking the backs of two other cabinets:
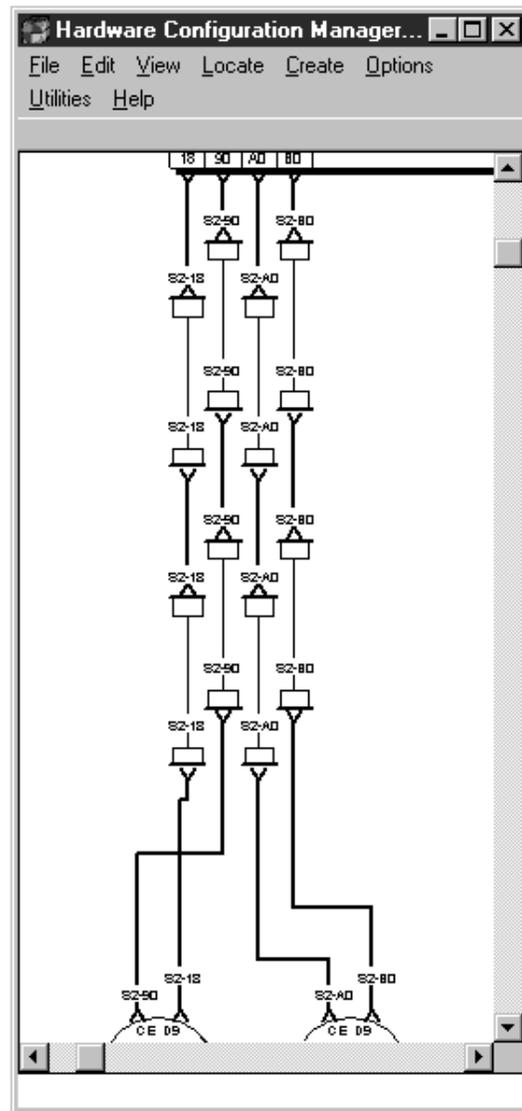
Figure 123. Configuration Diagram

In such a configuration, all of the trunk connections would be marked as permanent to indicate that they will not change. If you wanted to disconnect one of the ports, say D9 (perhaps so it could be connected later to another CHPID), HCM would detect that there are two cable positions at which the connection could be broken (that is, the two jumper connections linking the fronts of cabinets) and ask you to choose one of those two positions as follows:
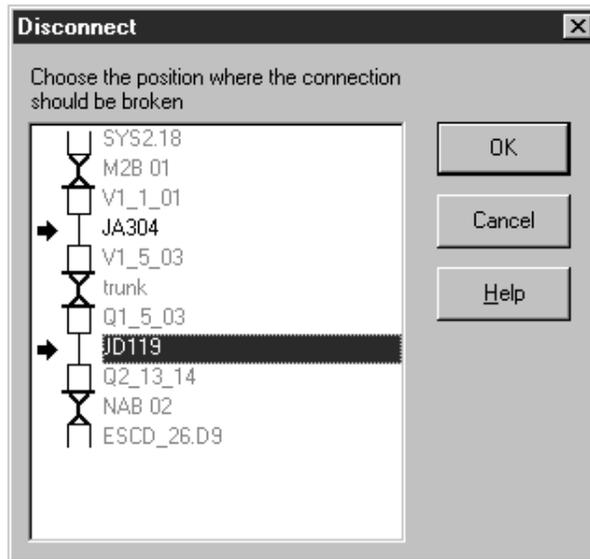
Figure 124. Disconnect Dialog

This dialog indicates the mutable connections by an arrow and preselects the lowest mutable connection. Clicking OK or double-clicking a mutable link will break the connection.

If you choose to break the connection in the lower position (using cable JD119), the other jumper cable (JA304, in this case) will remain in place even though it is not a permanent connection. This means that if you later connect, say, a switch port to CHPID SYS2.18, patchport Q1_5_03 and jumper JA304 will automatically be included in the fiber link of the new connection.

This dialog will not appear if there is only one mutable connection (e.g., there is only one jumper connecting the CHPID and port); in such cases, the connection will be broken at the one mutable link. Also, if the connection involves a P->S or S->P converter, HCM will automatically disconnect in such a way as to remove the converter and leave all the other connections intact. Finally, if all the links in the chain are marked as permanent, HCM will refuse to break the connection.

Connecting crossbar switches

Crossbar switches can be inserted into the connection between controllers and processors. The crossbar *channel interface* can connect upwards to either a processor CHPID or another crossbar's *control unit interface*. A crossbar may also connect upwards to a switch (through a converter), but not vice versa. That is, a crossbar channel interface can connect to a switch port, but a switch port may not connect to a crossbar control unit interface. Crossbar interfaces are parallel (bus and tag), unlike switch interfaces, which are serial (ESCON or FICON). Thus crossbars are never part of fiber links.

To insert a crossbar switch into an existing connection, you must first disconnect the controller from the processor. Then connect the crossbar to the processor. Finally, connect the controller upwards to the crossbar (see "Connecting controllers to processors and switches" on page 163 for more information).

Connecting a crossbar switch

To connect a crossbar switch, open the **Crossbar Switch** dialog by double-clicking on a crossbar or choosing **Crossbar Switch...** from the **Edit** menu. Select a crossbar channel interface and press the **Connect...** button. The **Connect Channel Interface** dialog (similar to the one used to connect controller channel interfaces) opens to allow you to select a target interface.

Select a processor CHPID, another crossbar or a switch in the list box. You can double-click on a target object with a plus (+) sign to expand it to display its interfaces; double-click on a target with a minus (-) sign to contract it. After you have selected the target, press the **OK** button. Alternatively, double-click on the target interface.

Positioning a crossbar switch on a daisy-chain

If you are connecting the crossbar channel interface to a CHPID which is already connected to another crossbar switch, the **Position Interface** dialog appears for you to position the crossbar in the connection. Select a crossbar in the Channel Interfaces list box and drag it to the desired position in the list. See "Positioning objects along daisy-chained connections" on page 156 for more details.

Disconnecting a crossbar channel interface

Open the **Edit Crossbar Switch** dialog by double-clicking on a crossbar in the configuration diagram. Select the connected crossbar channel interface from the Channel Interfaces list box and press the **Disconnect** button. HCM will ask you to confirm the disconnection before doing it.

Chapter 9. Utilities

HCM offers several utilities to assist in Fibre Channel migration and other time-consuming tasks. These utilities support you in making complex configuration changes without violating any logical definitions that are valid in your configuration.

You invoke the available utilities from the **Utilities** menu. They are wizards that guide you through the various functions. After changing your configuration, you will be shown a summary of your changes and asked whether you wish to accept or discard them. The summary can also be saved to a file.

> **CHPID Mapping Tool Support**
>
> The **CHPID Mapping Tool Support** utility, that you can also invoke from the **Utilities** menu, is described in "How to use the CHPID Mapping Tool Support utility" on page 84.

Moving port connections

This utility allows moving port connections. In particular, you can move complete CHPID-to-control-unit connections that have a path through an ESCON or FICON switch from one switch to another, or rearranging those connections on a single switch. Using this utility, you no longer have to manually disconnect and connect the involved objects.

When moving port connections using the utility, the following data will be automatically updated:
- Link address of affected control units
- Dynamic switch parameter of the affected CHPIDs
- Port configurations of source and target port when moving within the same switch

You can only move ports that are connected to another object.

When you move a port connection to another switch, note that all other connections that are *implied* by this move are also moved. That is, in addition to the selected CHPID-to-control-unit connection, HCM also moves all connections that are required to keep the logical definitions between CHPIDs and control units valid. For example, moving a CHPID from switch A to switch B implies a move of the control unit connections that are serviced by that CHPID and, furthermore, all CHPIDs that have access to those control units via switch A. See "Implied ports" on page 202 for an example of implied ports.

Since a move of one port connection may imply a move of several other port connections, be aware that there might not be enough empty ports available on the target switch.

You cannot move port FE if this port is the switch control unit port of the switch of interest. Therefore, if port FE is implied in a port move, the move cannot be performed. You first have to disconnect port FE by disconnecting the internal control unit from the related CHPIDs.

If you move a port connection on the same switch, all port matrices are updated accordingly. If you move a port to another switch neither the target port matrices nor the source port matrices are updated. You have to manually update your source and target configurations.

If a port or connection, which is involved in the move, has a logical or physical inconsistency, the move will not be allowed. For example, a control unit has a link address to a port but physically the control unit is not available. If this is the case, a move that involves this port is not allowed. See "Receiving visual warnings" on page 57 for more information on physical inconsistencies.

Implied ports

The following example explains that moving a port connection might imply that other port connections also need to be moved. Assume, you want to move CHPID 21 that connects to port C1 to another switch in the following example.
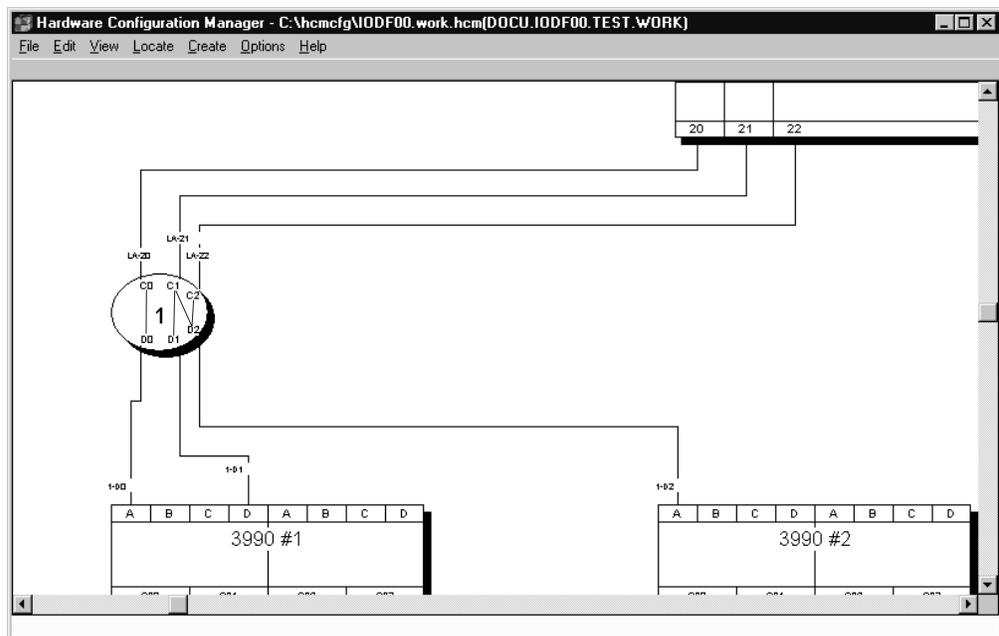
Figure 125. Example for Implied Ports

Assume a CHPID 21 connected to a port C1 has a path to two control units that are connected to port D1 and D2. Furthermore, the control unit of port D2 is also serviced by a CHPID 22 connected to port C2. When you want to move port C1 with its connections, the utility will also automatically move ports D1, D2, and C2 to the target switch.

Procedure

The utility guides you through the following steps:

1. Select **Move Port Connections...** from the **Utilities** menu.
2. Select a switch as target.

3. Choose one or more ports whose connections you want to move to the target switch. The ports may be from any switch, including the target switch.
4. Choose ports on the target switch for all ports to be moved.
5. Before you finish the moving of port connections, a list shows you all changes that will be made. Save this list and finish the move.

Aggregating CHPIDs

This utility allows you to combine attachments of several CHPIDs onto a single CHPID. That is, all control units connected to the source CHPIDs will be disconnected from the source CHPIDs and connected to a target CHPID.

All the CHPIDs should be connected to the same switch and use the same dynamic switch.

When you aggregate CHPIDs, HCM does not change the reachability of devices by logical partitions (LPARs). The LPARs in the access or candidate list of the source CHPID must be a subset or equal the LPARs in the access or candidate list of the target CHPID. If the target CHPID is shared and the LPARs in the access or candidate list of the CHPIDs to be aggregated are a subset of the LPARs in the access or candidate list of the target CHPID, then an explicit device candidate list will be built. When the device already had an explicit device candidate list, it will be left unchanged.

If a CHPID was involved in a CTC connection, all CTC control units connected to the CHPID entry port are moved to the CHPID entry port of the target switch. The CU link addresses are updated accordingly to create valid CTC connections.

The utility guides you through the following steps:
1. Select **Aggregate CHPIDs...** from the **Utilities** menu.
2. Choose a CHPID as the target and press **Next**.
3. Choose one or more CHPIDs to aggregate with the target CHPID.
4. Before you finish the aggregation of CHPIDs, a list shows you all changes that will be made. Save this list and finish the aggregation.

Converting CNC channels to FICON channels

This utility supports you in making the necessary I/O configuration changes when you install a new FICON bridge card on an ESCON switch. Before you can start a conversion, the following prerequisites must be met:

- You can only convert a CHPID that is already connected to a port and defined as type CNC or FCV. If the CHPID is CNC, it must be eligible to be an FCV channel. HCM automatically performs a CHPID type change.
- The ESCON port that connects to the CHPID must be FCV eligible.

HCM checks whether the correct number of unconnected port addresses follow the ESCON port that connects to the new channel. If the installation of the new card causes a conflict with already connected ports, HCM initiates moving of the affected ports or aggregating CHPID-to-control-unit definitions. Refer to the description of the wizards "Moving port connections" on page 201 and "Aggregating CHPIDs" on page 203 for considerations on those subjects.

The utility guides you through the following steps:
1. Select **Convert ESCON Port to FC Port...** from the **Utilities** menu.

2. Choose an ESCON port that is FCV eligible and already connected to a CNC or FCV channel.

3. A dialog appears that lets you aggregate source CHPIDs to a target CHPID. Refer to "Aggregating CHPIDs" on page 203 for an example on things to consider when aggregating CHPIDs.

4. If some of the switch ports displaced by the fiber channel port were previously connected, you are prompted to move the required port. Refer to "Moving port connections" on page 201 for an example on things to consider when moving ports.

5. Before you finish the conversion, a list shows you all changes that will be made. Save this list and finish the conversion.

See "Scenario IV – Configuration changes for installing a FICON bridge card in a 9032–5 ESCON director" on page 303 for an example of the use of this utility.

Priming the IODF with sensed data

This utility allows you to save sensed data such as serial numbers and volume serial numbers (VOLSER) into the IODF. Before you can start working with this wizard, data must first be sensed. To do this, use the 'Sense' command from the Operations menu to populate the configuration with sensed data. A prerequisite for this wizard is the availability of the I/O Operations functions; otherwise, sensing of data is not possible.

> **Note:**
> The I/O Operations functions are not available if your system is running as a host under z/VM.

The utility guides you through the following steps:

1. Choose the set of IODF serial numbers (on processors, switches, control units and devices) that will be updated with the sensed serial number value.

2. Choose the set of IODF VOLSERs that will be updated with the sensed VOLSER value.

3. Examine a summary of the sensed data that is to be updated. Before you finish the utility, you can save the list of changes. Click 'Finish' to perform the operation you selected in the previous steps.

Data which has been sensed can be deleted later by selecting **Clear Previous Sensing Data** in the **Sense** submenu of the **Operations** menu.

Changing the CHPID link address format

This utility allows you to modify the link address format for FC channel paths. Starting with a specific processor support level for the 2064 processor family, the channel paths support the definition of two–byte link addresses. Using this utility, you can explicitly upgrade one–byte link addresses for a selected FC channel path to two–byte link addresses. Similarly, you can also downgrade two–byte link addresses for a selected FC channel path to one–byte link addresses. Remember that no link address format mix is allowed for FC channel paths, that means all connections are defined either via one–byte link addresses or two–byte link addresses.

The utility guides you through the following steps:

1. Choose whether to convert one–byte link addresses into two–byte link addresses or vice versa.
2. Select the CHPID(s) for which the link address format has to be converted.
3. Examine the summary of modified link addresses.

Before you finish the utility, you can save the list of changes. Click **Finish** to perform the operation you selected in the previous steps.

Creating an I/O subsystem

This utility allows to create an I/O subsystem consisting of

- controllers (with segments and control units) and
- strings (with devices).

The boundaries and contents of the subsystem are based on selections of physical description files (PDF). You select one PDF to define the number of controllers (and their type), and another PDF to define the type of strings.

The utility guides you through the following steps:

1. Select **Create Subsystem...** from the **Utilities** menu. In the dialog that opens, select the controller physical definition file, number of controllers in the subsystem, and label prefix for controllers.
2. Add control units. Attributes for control units consist of type, number, subsystem-ID, offset between added CUs, serial number, and description.
3. Attach control units to controllers (and their segments).
4. Define strings. For this purpose select a string PDF (based on a selected type). Attributes to set are number of devices inside string, starting device number, label prefix, and description.
5. Connect strings and controllers (1): connect string interfaces and a controller device interfaces. Selectable interfaces are based on the settings made in steps 1 through 4.
6. Connect strings and controllers (2): connect strings and control units. Selectable objects are based on the settings made in steps 4 and 5.

Chapter 10. Operating system configurations

Each device can be defined to one or more operating systems. The hardware configuration contains this definition information, known as the *operating system (OS) configuration.* Note that you may define devices and their associated parameters to an OS configuration, even if they are not explicitly connected.

Typically, the OS configuration is associated with a processor or partition. Each processor can run different OS configurations at different times. There are two types of OS configuration: MVS and VM.

An MVS OS configuration can contain one or more EDTs. An *Eligible Device Table or EDT* is a list of esoteric device groups, plus the names of the generic device types which determine how devices are used (order of preference) within each esoteric. The EDT is identified by a two-digit ID; you must define at least one EDT for each MVS OS configuration.

An *esoteric device group* or *esoteric* is an installation-defined and named grouping of I/O devices under an EDT. An esoteric acts as a single "virtual device" which can be allocated without using a specific device number. Devices within an esoteric are usually from the same device group, but can be of different device types.

Every device type is associated with a *generic device group* or *generic*. A generic is an MVS-defined grouping of devices with similar characteristics, which determines how devices are used in an esoteric device group. Each generic has a *preference number* which governs the order of allocation of devices to jobs within an esoteric. For example, if the esoteric contains the device types 3390, 3380 and 3350, and the generics associated with the device types have preference numbers 280, 290 and 300, respectively, then the preference numbers are used by the MVS allocation component to assign the devices in that order of preference.

Faster devices have lower preference numbers. You can change the preference number of a generic under an EDT (*user-defined generic*), but not under an OS configuration (*system-defined generic*).

You can mark one or more of the devices defined to an OS configuration to be your consoles. A *console* displays system-related information (for example, error messages) directly from MVS. For the MVS system, you can specify NIP consoles; for z/VM, you can specify primary and alternate VM consoles.

> **Note:**
> Under z/VM, you can only view z/OS IODFs, but you cannot define z/OS operation system configurations.

Listing the OS configuration-EDT-esoteric hierarchy

Open the **Operating System Configuration** dialog by choosing **OS configurations...** from the **Edit** menu. The list box contains all OS configurations, EDTs and esoterics in a hierarchy in alphabetical order. You can double-click on an OS configuration (MVS only) with a plus (+) sign to display its EDTs; double-click

on an EDT with a plus (+) sign to display its esoteric device groups. Double-click on an item with a minus (-) sign to close it. Double-clicking on an esoteric opens the **List Devices for Esoteric** dialog.

The static text fields below the list box display information for the item currently selected in the list box. If an OS configuration is selected, its name, type and description are displayed. If an EDT is selected, its ID, parent OS configuration, description, time of last update, and updating user name are displayed. If an esoteric is selected, its name, parent OS configuration, state, token, and VIO-eligibility are displayed. The name of an esoteric follows the EE.SSSSSSSS format, where EE is the EDT name and SSSSSSSS is the esoteric name. The state of an esoteric can be one of the following:
- *no devices*
- *has devices*
- *no-DASD devices but VIO=YES*

You can access almost every dialog dealing with OS configuration from the **Operating System Configuration** dialog. You can create or edit OS configurations, EDTs or esoterics, as well as list and/or edit generics, consoles or the devices defined to the OS configuration or esoteric.
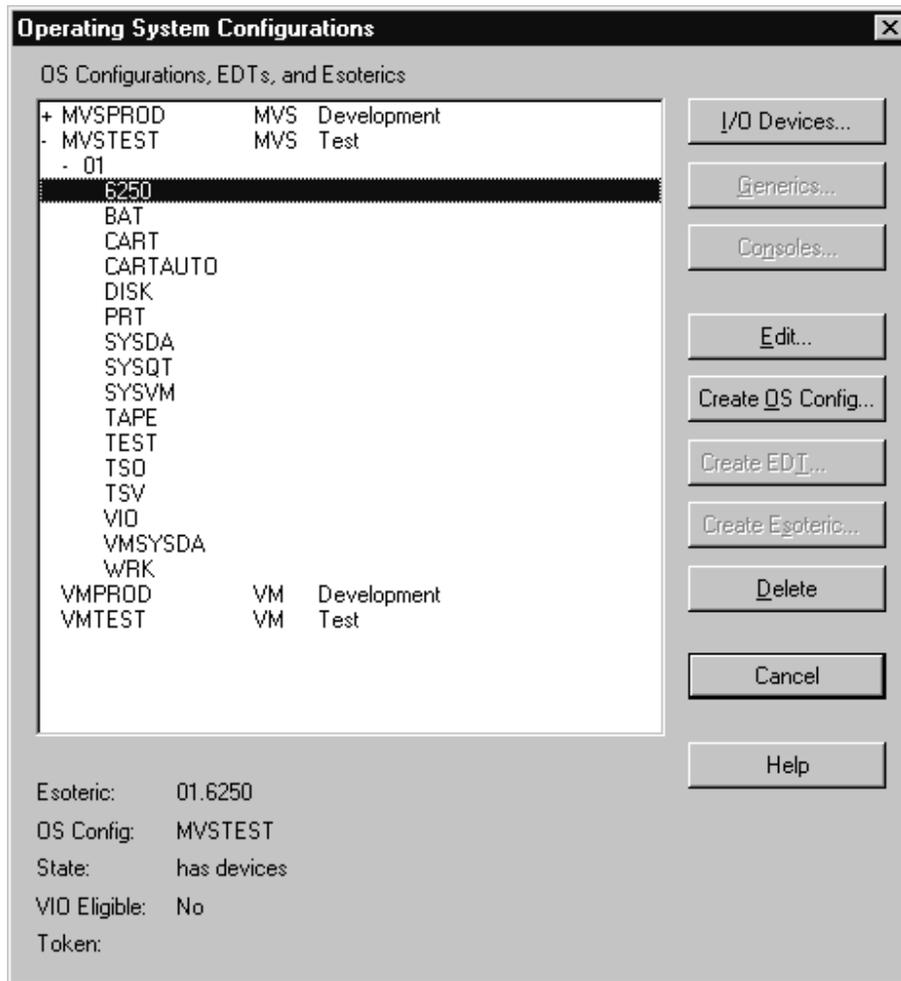
Figure 126. Operating System Configurations Dialog

Creating an OS configuration

Open the **Create Operating System Configuration** dialog by pressing the **Create OS Config...** button in the **Operating System Configurations** dialog. Fill in the name, type and description of the OS configuration and press **OK**; the new OS configuration will appear in the Operating System Configurations dialog box. After you have created an MVS OS configuration, you can add EDTs and esoteric device groups under it, as well as define devices to it.
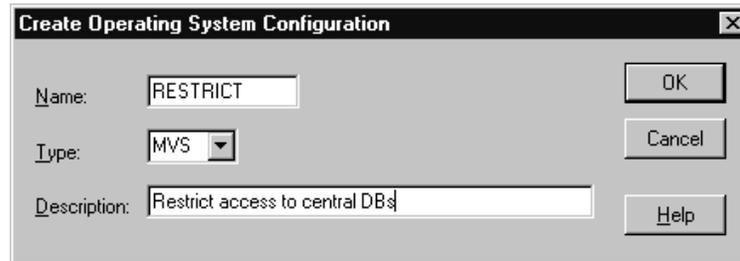
Figure 127. Create Operating System Configuration Dialog

Editing an OS configuration

Select an OS configuration and press the **Edit...** button in the **Operating System Configurations** dialog. The **Edit Operating System Configuration** dialog appears. Edit the description and press **OK**.

Deleting an OS configuration

Select the OS configuration in the list box and press the **Delete** button in the **Operating System Configurations** dialog. HCM will display a confirmation dialog before deleting the OS configuration.

Listing devices for an OS configuration

Open the **List Devices for OS Configuration** dialog by selecting an OS configuration and pressing the **I/O Devices...** button in the **Operating System Configurations** dialog. The Devices list box in this dialog displays all the devices defined for the selected OS configuration. Groups of consecutive device numbers appear on one line each with a plus (+) sign. Double-click on a device range with a plus (+) sign to expand it to display each device on its own line. Double- click on a device range with a minus (-) sign to contract it. Note that the information on a device group line apply only to the first device in the group. It is possible, therefore, that other devices in the group may have column values that differ from the first device. To see the column values for individual devices, these entries must be expanded.

You can display the parameters, attached control units, serial numbers or descriptions for the devices by choosing the appropriate View option. If you choose the Parameters view radio button, you can select which parameters to display from the **Show Parameters / Features as Columns** dialog. Open this dialog by pressing the **Columns...** button. Add or remove parameters or features to or from the Exclude and Include list boxes. Default parameters displayed for MVS operating systems are "DYNAMIC" and "OFFLINE", for VM operating systems, "UIRATE" and "OFFLINE".

You can also choose to restrict the number of devices in the Devices list box by clicking in the Use Filter check box, and then selecting a set of filters from the

Filter I/O Device List dialog (press the **Filter...** button to open the dialog). Devices which appear in the list box match all the filtering criteria that you set in this dialog. You can set the device type, class, serial number or description; connected, unconnected or specific control units; and/or the processor or partition. The Parameters / Features section at the bottom of the filtering dialog shows only those parameters or features selected in the **Show Parameters / Features** dialog which you can use as filtering criteria. (The Parameters / Features section appears only when the filtering dialog is called from the **List Devices for an OS Configuration** dialog.)
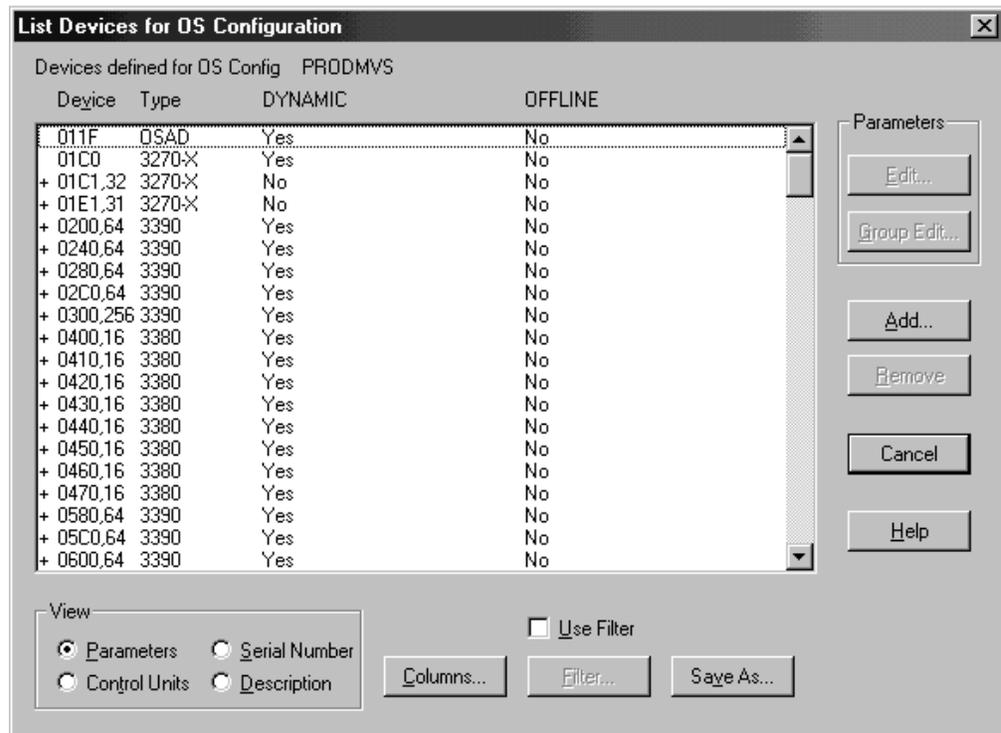
Figure 128. List Devices for OS Configuration Dialog

Note: Using the **Save As...** button, you can save the list of devices into a text file. This saved file contains all information about the devices, whereas in the dialog, at most five device parameters are visible at a time.

Adding devices to an OS configuration

You can define devices to the selected OS configuration. In the **List Devices for an OS Configuration** dialog, press the **Add...** button to open the **Add I/O Devices to OS Configuration** dialog.

The list box in this dialog shows all devices defined in the IODF which are not currently defined to the selected OS configuration, and which match the current filter (if enabled). Groups of consecutive device numbers appear on one line each with a plus (+) sign. Double-click on a device range with a plus (+) sign to expand it to display each device on its own line. Double-click on a device range with a minus (-) sign to contract it. Note that the information on a device group line apply only to the first device in the group. It is possible, therefore, that other devices in the group may have column values that differ from the first device. To see the column values for individual devices, these entries must be expanded.

Select one or more devices and press **OK** to add them to the OS configuration. (Alternatively, double-click on the device.) The **Define Device <=> OS Configuration Parameters** dialog automatically appears, to allow you to update the parameters defining the connection between the device and the OS configuration. If you have selected more than one device, the parameters dialog appears only once, but any changes made to parameters apply to all selected devices. For more details, see "Editing device <=> OS configuration parameters" on page 212.

You can determine what device information appears in the list box by selecting one of the View options: connected control units, device descriptions or serial numbers. You can also choose to restrict the number of devices which appear in the list box by using the filtering option (see "Listing devices for an OS configuration" on page 209 for more details on filtering).

Note that an Operating Systems section appears at the bottom of the **Filter I/O Device List** dialog only when the dialog is called from the **Add I/O Devices to OS Configurations** dialog. If you select the *No OS Filtering* option, all devices match and are displayed in the list box; if you select *Not Defined*, a device will match the criterion if it is *not* defined to any OS configuration; if you select *Defined*, a device will match if it is defined to any OS configuration.
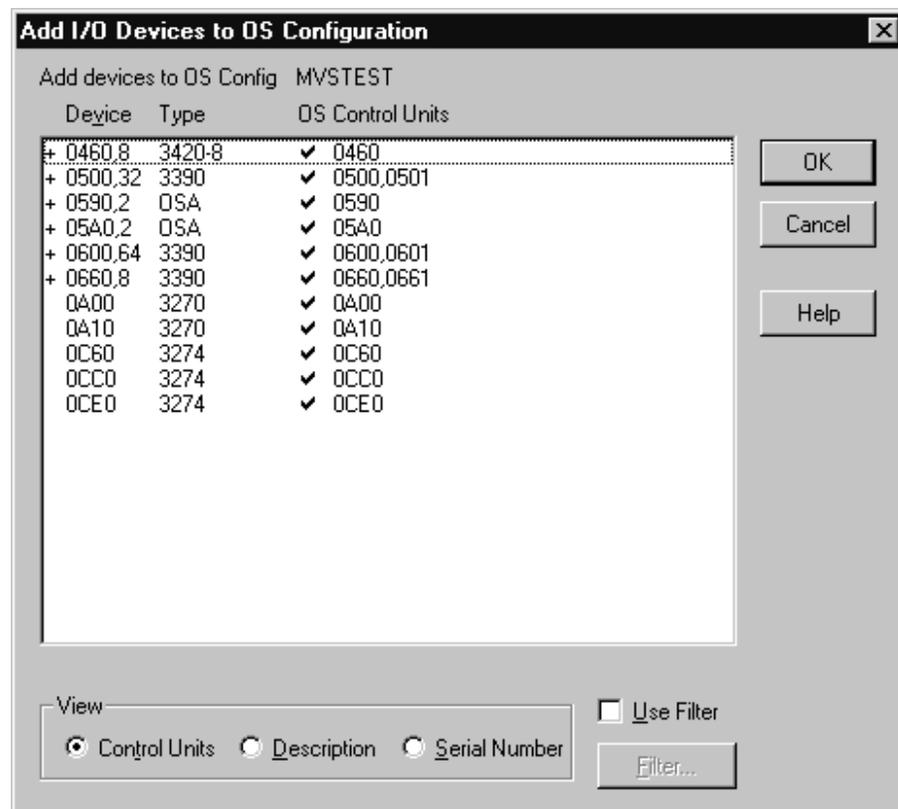
Figure 129. Add I/O Devices to OS Configuration Dialog

Removing devices from an OS configuration

In the **List Devices for OS Configuration** dialog, select one or more devices and press the **Remove** button. HCM will display a confirmation dialog before disconnecting the selected device(s) from the OS configuration.

Editing device <=> OS configuration parameters

Select one or more devices from the list box in the **List Devices for an OS Configuration** dialog, and press either the **Edit...** or **Group Edit...** button to open the **Define Device <=> OS Configuration Parameters** dialog. You can then modify the parameters which define the selected device(s) to the OS configuration. If you selected more than one device and pressed **Edit...**, the parameters dialog appears repeatedly, once for each device. If you selected more than one device and pressed **Group Edit...**, the dialog appears only once, displaying the parameters for the first selected device. Any changes affect all selected devices. HCM will display an error message if the selected devices are incompatible for a group edit (that is, they are of different types).
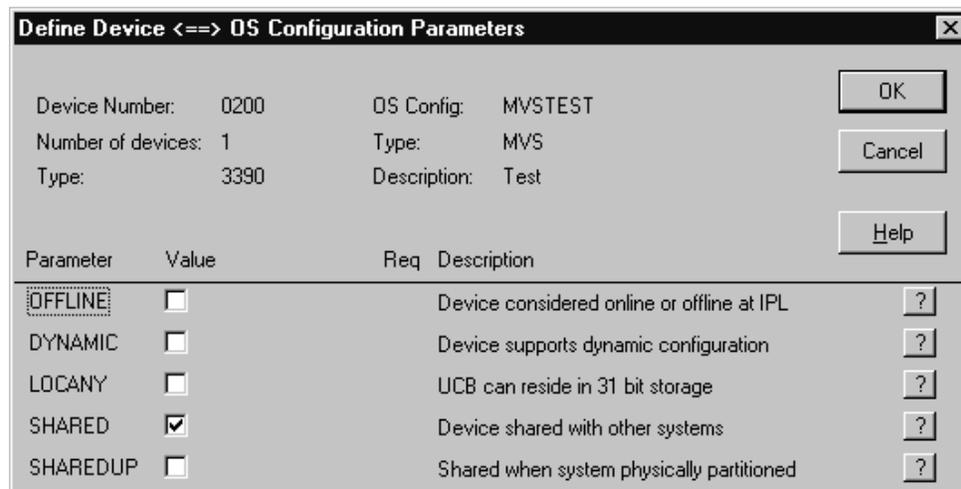
Figure 130. Define Device <=> OS Configuration Parameters Dialog

You can also reach the parameters dialog from the **Add I/O Devices to OS Configuration** dialog or the **List Device <=> OS Configuration Definitions** dialog. See "Adding devices to an OS configuration" on page 210 or "Selecting OS configurations to which to define devices" on page 213 for more details.

The static text fields at the top of the **Define Device <=> OS Configuration Parameters** dialog identify the device and OS configuration involved in the connection. In the Parameters section below, each row of information contains the parameter name, value, whether it is required, and a description. You can update the value of a parameter by selecting one from the drop down box, typing in a text box, or clicking in a check box, as appropriate. To get a more detailed explanation of a parameter's meaning, click on the **"?"** button on its row.

Listing OS configurations to which devices are defined

You can select one or more devices and list the OS configurations to which those devices are defined. Open the **I/O Device List** dialog by clicking on the **I/O Devices...** button in the **String** dialog to display all devices for a selected string. Alternatively, click on the **I/O Devices...** button in the **Control Unit List** dialog (which is called from the **Controller** dialog) to display all devices for a selected control unit.

Select a device(s) from the **I/O Device List** dialog and press the **OS Configs...** button to open the **List Device <=> OS Configuration Definitions** dialog. The list box in this dialog displays all OS configurations available in the IODF. A check mark under the "Defined" column indicates the selected device is defined to the

OS configuration. If more than one device is selected, the check mark only applies to the first device.
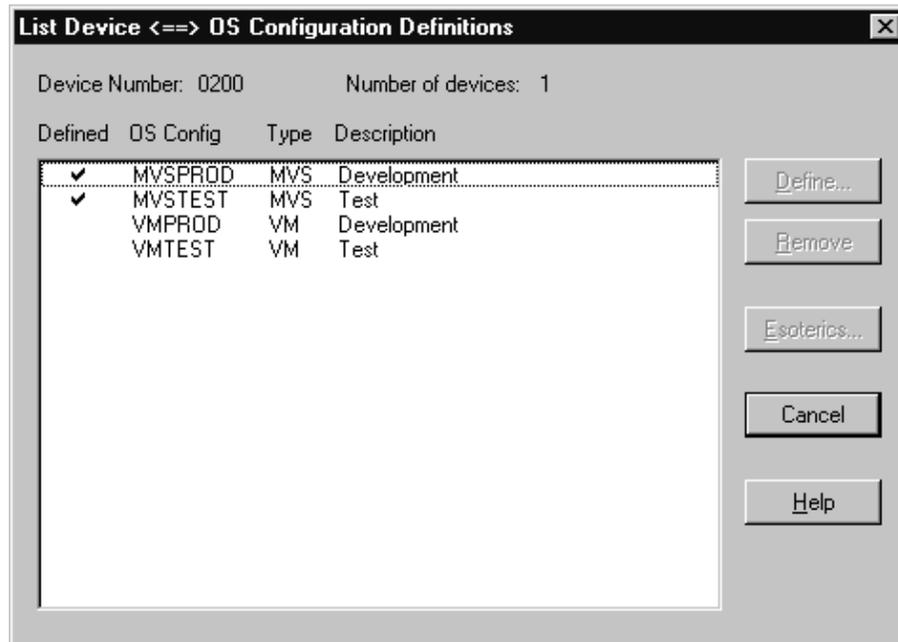
Figure 131. List Device <=> OS Configuration Definitions Dialog

Selecting OS configurations to which to define devices

You can select one or more OS configurations in the list box to define to the selected device(s). Press the **Define...** button to open the **Define Device <=> OS Configuration Parameters** dialog. You can then update the parameters determining the device <=> OS configuration definition. See "Editing device <=> OS configuration parameters" on page 212 for more details on this dialog.

Selecting OS configurations from which to remove devices

Select one or more OS configurations and press the **Remove** button. HCM will display a confirmation dialog before removing the devices selected in the **I/O Device List** dialog from all selected OS configurations.

Adding devices to an esoteric within the OS configuration

Select an OS configuration(s) and press the **Esoterics...** button to open the **Include Devices in Esoterics** dialog. In this dialog, you can include (or exclude) the devices selected in the I/O Device List in esoteric device groups within the selected OS configuration. If you selected more than one OS configuration, the esoterics dialog appears repeatedly, once for each selection.
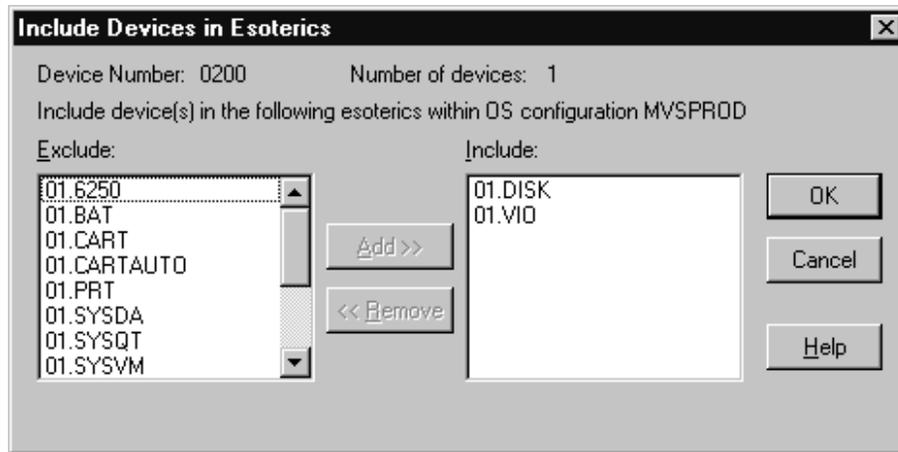
Figure 132. Include Devices in Esoterics Dialog

Listing consoles for an OS configuration

Open the **List Consoles** dialog by selecting an OS configuration in the **Operating System Configurations** dialog and pressing the **Consoles...** button. The list box displays all the devices which have been marked as consoles for the selected OS configuration, organized by order number.
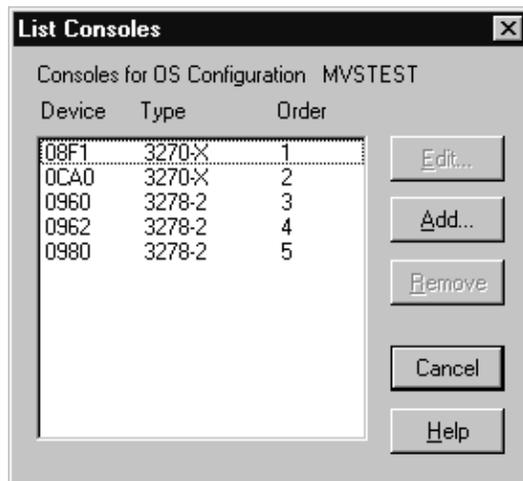
Figure 133. List Consoles Dialog

Adding a console to an OS configuration

Press the **Add...** button in the **List Consoles** dialog to open the **Add Console** dialog. You can then select a device to define as a console for the selected OS configuration. The drop down box displays all devices defined to the OS configuration which are eligible to be used as consoles.
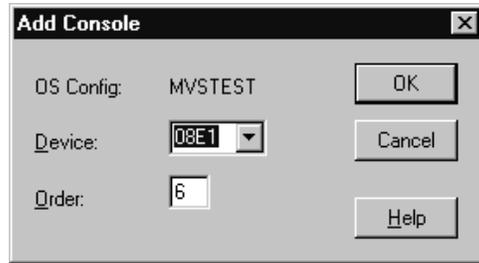
Figure 134. Add Console Dialog

Removing a console from an OS configuration

Select the console and press the **Remove** button in the **List Consoles** dialog. HCM will display a confirmation dialog, then remove the console from the list box. The device is no longer marked as a console, but is still defined to the OS configuration.

Editing a console for an OS configuration

Select a console from the **List Consoles** dialog and press the **Edit...** button. The **Edit Console** dialog appears to allow you to edit the order number for the selected console.
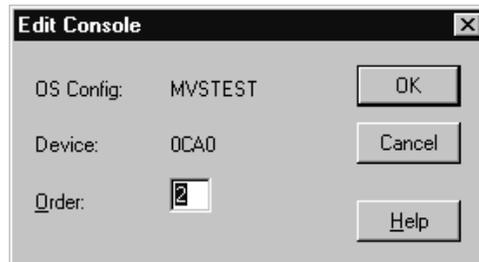
Figure 135. Edit Console Dialog

Eligible device tables (EDTs)

An MVS OS configuration can contain one or more EDTs. An Eligible Device Table or EDT is a list of esoteric device groups, plus the names of the generic device types which determine how devices are used (order of preference) within each esoteric. The EDT is identified by a two-digit ID; you must define at least one EDT for each MVS OS configuration.

Creating an EDT

After you have created an OS configuration, you can create EDTs or Eligible Device Tables under it. Select an OS configuration (MVS only) and press the **Create EDT...** button in the **Operating System Configurations** dialog. The **Create Eligible Device Table** dialog opens, allowing you to specify an EDT name and description. After you press **OK**, the new EDT will appear in the list box of the **Operating System Configurations** dialog, under the selected OS configuration.
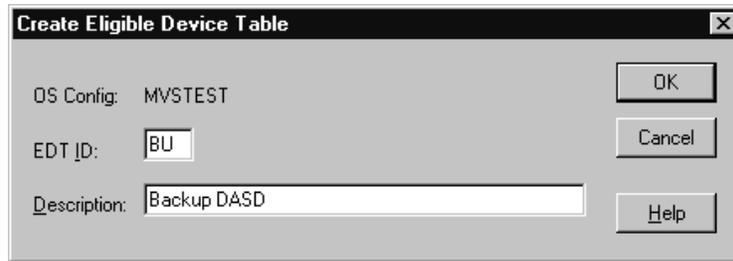
Figure 136. Create Eligible Device Table Dialog

Editing an EDT

Select an EDT from the list box in the **Operating System Configurations** dialog and press the **Edit...** button. The **Edit Eligible Device Table** dialog appears, allowing you to edit the EDT description.

Deleting an EDT

Select the EDT in the list box and press the **Delete** button in the **Operating System Configurations** dialog. HCM will display a confirmation dialog before deleting the EDT.

Esoteric device groups (esoterics)

An *esoteric device group* or *esoteric* is an installation-defined and named grouping of I/O devices under an EDT. An esoteric acts as a single "virtual device" which can be allocated without using a specific device number. Devices within an esoteric are usually from the same device group, but can be of different device types.

Creating an esoteric

After you have created an OS configuration and EDT, you can add esoteric device groups. Open the **Create Esoteric Device Group** dialog by selecting an EDT and pressing the **Create Esoteric...** button in the **Operating System Configurations** dialog. Fill in the name, token and VIO eligibility. After you press **OK**, the new esoteric will appear in the list box of the **Operating System Configurations** dialog, under the selected OS configuration and EDT.
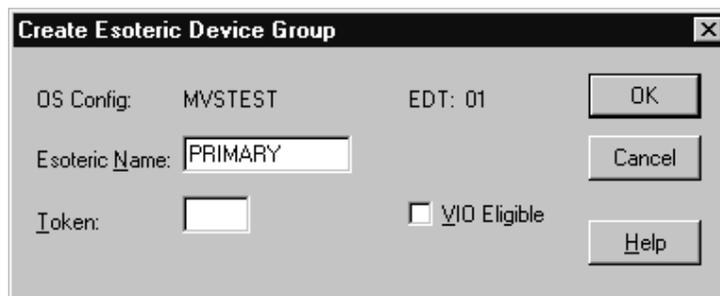
Figure 137. Create Esoteric Device Group Dialog

Editing an esoteric

Select an esoteric from the list box in the **Operating System Configurations** dialog and press the **Edit...** button. The **Edit Esoteric Device Group** dialog appears, allowing you to edit the esoteric token and VIO eligibility.

Deleting an esoteric

Select the esoteric in the list box and press the **Delete** button in the **Operating System Configurations** dialog. HCM will display a confirmation dialog before deleting the esoteric.

Listing devices for an esoteric

Select an esoteric and press the **I/O Devices...** button in the **Operating System Configurations** dialog. The **List Devices for Esoteric** dialog appears, allowing you to examine the devices defined for the selected esoteric. List box entries may be either a single device or range of devices. Range entries use the format XXXX,YY where XXXX is the first device number and YY indicates the number of devices in the range. Double-click on a range with a plus (+) sign to expand it to display each device on its own line. This allows you to delete some but not all of the devices in the range. Double-click on a range with a minus (-) sign to collapse it.
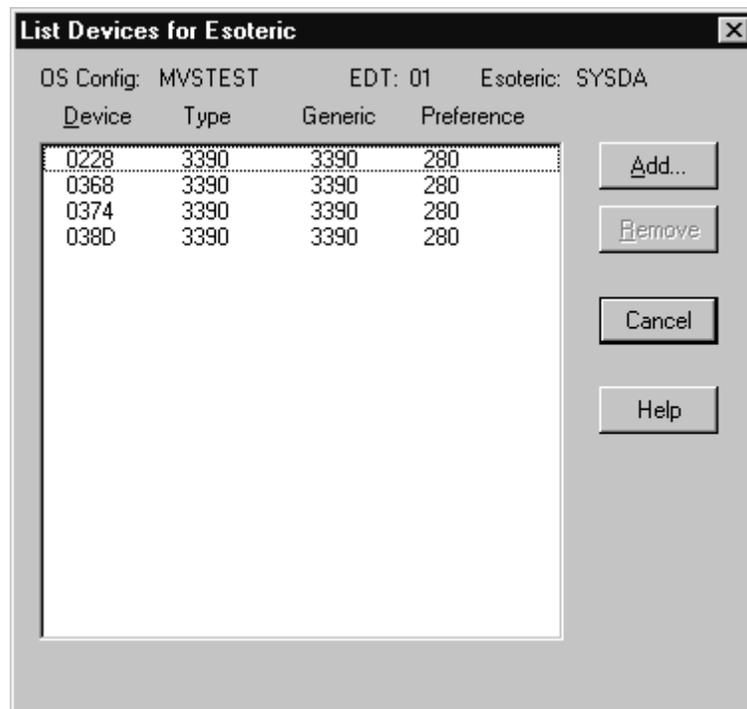
Figure 138. List Devices for Esoteric Dialog

Adding devices to an esoteric

Open the **Add I/O Devices to Esoteric** dialog by pressing the **Add...** button in the **List Devices for Esoteric** dialog. You can then select one or more devices from the list box to add to the esoteric. The list box displays all devices currently defined to the OS configuration which contains the esoteric, and which match the current filter (if enabled). Groups of consecutive device numbers appear on one line each with a plus (+) sign. Double-click on a device range with a plus (+) sign to expand it to display each device on its own line. Double-click on a device range with a minus (-) sign to contract it. Note that the information on a device group line

apply only to the first device in the group. It is possible, therefore, that other devices in the group may have column values that differ from the first device. To see the column values for individual devices, these entries must be expanded.

You can determine what device information appears in the list box by selecting one of the View options: connected control units, device descriptions or serial numbers. You can also choose to restrict the number of devices which appear in the list box by using the filtering option (see "Listing devices for an OS configuration" on page 209 for more details on filtering).
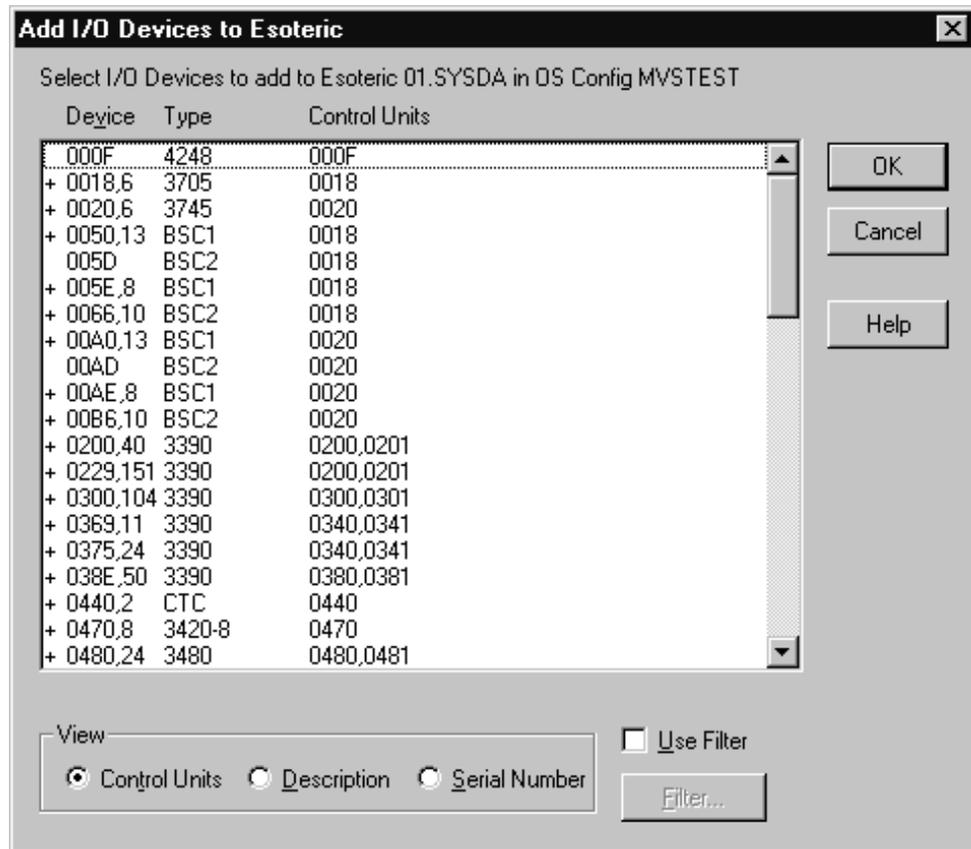
Figure 139. Add I/O Devices to Esoteric Dialog

Removing devices from an esoteric

Select one or more devices and press the **Remove** button in the **List Devices for Esoteric** dialog. HCM will display a confirmation dialog before disconnecting the selected device(s) from the esoteric.

Generic device groups (generics)

Every device type is associated with a *generic device group* or *generic*. A generic is an MVS-defined grouping of devices with similar characteristics, which determines how devices are used in an esoteric device group. Each generic has a *preference number* which governs the order of allocation of devices to jobs within an esoteric. For example, if the esoteric contains the device types 3990, 3880 and 3850, and the generics associated with the device types have preference numbers 280, 290, and 300, respectively, then the preference numbers are used by the MVS allocation component to assign the devices in that order of preference.

Faster devices have lower preference numbers. You can change the preference number of a generic under an EDT (*user-defined generic*), but not under an OS configuration (*system-defined generic*).

Listing system-defined generics

Open the **List System-defined Generics** dialog by selecting an MVS OS configuration and pressing the **Generics...** button in the **Operating System Configurations** dialog. You can then browse the list of generic device groups defined for the selected OS configuration. You can order the list by name or by preference by choosing the appropriate View option.

Note that you cannot edit system-defined generics (under an OS configuration), but you can change the preference number for *user-defined generics* (under an EDT).
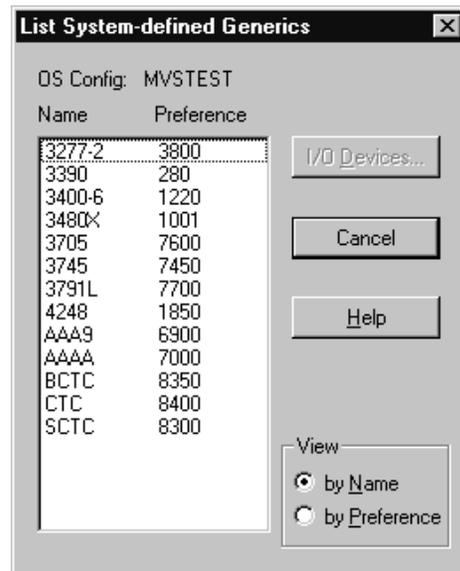
Figure 140. List System-Defined Generics Dialog

Listing devices for a system-defined generic

Select a generic and press the **I/O Devices...** button in the **List System-defined Generics** dialog. The **List Devices for Generic** dialog appears, allowing you to browse the list of devices associated with the generic.

Listing user-defined generics

Open the **List User-defined Generics** dialog by selecting an EDT and pressing the **Generics...** button in the **Operating System Configurations** dialog. You can then browse the list of generic device groups defined for the selected EDT. The list box displays the generics by name or by preference, depending on which View option you choose.

Note that you can edit *user-defined generics* (defined under an EDT), but not *system-defined generics* (under an OS configuration).
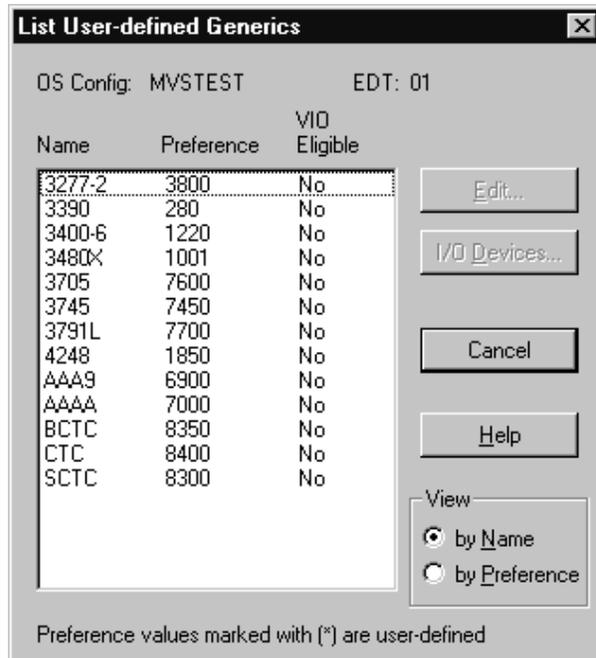
Figure 141. List User-Defined Generic Dialog

Listing devices for a user-defined generic

Select a generic and press the **I/O Devices...** button in the **List User-defined Generics** dialog. The **List Devices for Generic** dialog appears, allowing you to browse the list of devices associated with the generic.

Editing a user-defined generic

You can change the preference value and VIO eligibility of a user-defined generic in the **Edit Generic** dialog. To display this dialog, select a user-defined generic and press the **Edit...** button in the **List User-defined Generics** dialog. Note that leaving the preference number text box blank and pressing **OK** will reinstate the system-defined preference number for the generic.

Chapter 11. User interface to I/O Operations

HCM provides additional capabilities which allow I/O Operations tasks on the host to be performed via HCM. To be able to perform these functions, I/O Operations must be up and running. The following functions are available with HCM:

- Viewing the active sysplex and the status of its objects graphically,
- Retrieving, manipulating and activating switch matrices,
- Issuing I/O Operations commands.

To carry out these functions, you must, of course, have the necessary RACF authorization to perform I/O Operations (see *System Automation for z/OS User's Guide*). To view the status information of the active system, it is obvious that the active configuration must correspond with the HCM configuration. This is the responsibility of the user. If the HCM configuration does not represent the active IODF, misleading results will occur. Note that to manipulate an active switch matrix you require access to the active IODF.

To access the I/O Operations on the host, the prerequisites mentioned under "I/O Operations functions" on page 5 are necessary.

> **Note:**
> The I/O Operations functions are not available if your system is running as a host under z/VM.

Viewing the active sysplex and its status

HCM knows the systems of the active sysplex (that is, the sysplex on which HCD is running) and associates them with partitions and processors (for basic mode). Because of this, partitions and basic mode processors can be labelled with their associated system name in the configuration diagram according to the **Operations Preferences** you have selected.

To display all the objects of the active sysplex, use the filter option in the **View** menu and select the **Active sysplex** checkbox. The resulting diagram will consist of the desired systems and their objects.

To see the status of the objects in the sysplex, use the menu item **Status** under the **Operations, Refresh** menu items. HCM now retrieves the status information via I/O Operations and shows it on the diagram in various colors having different meanings which can be defined in the **Color** dialog (see "Color dialog" on page 224).

Detailed information for a selected object can be obtained by clicking on the object with the right mouse button and selecting **Status** in the upcoming popup menu.

If you do this for a Processor or Partition object, the following dialog appears.
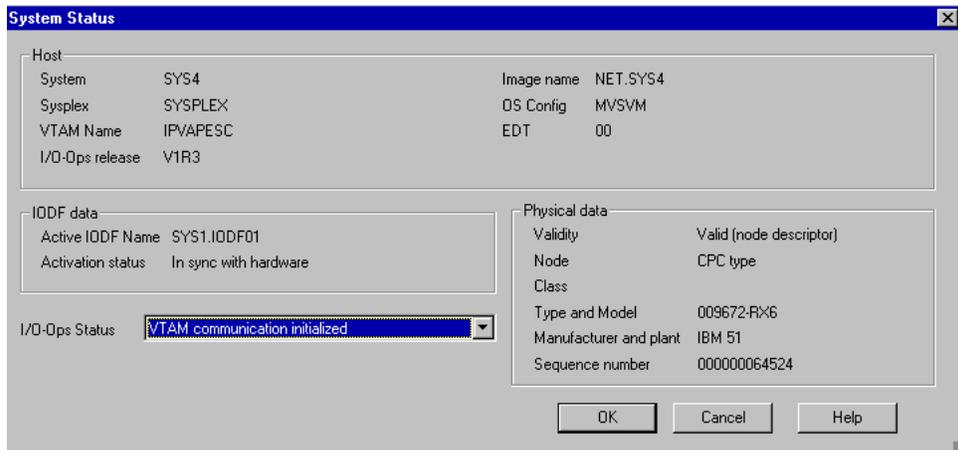
Figure 142. System Status dialog

The dialog shown in Figure 143 appears if you choose **Status...** from the popup menu for a string or a controller with integrated devices.
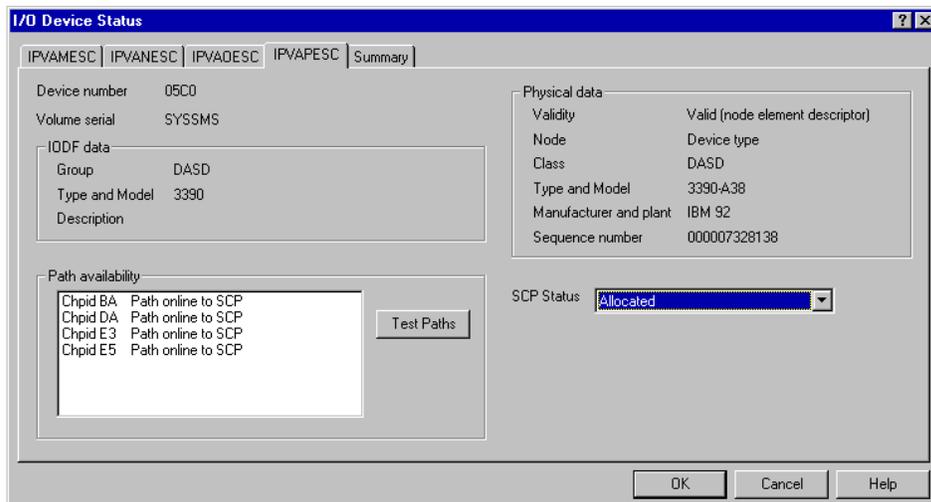
Figure 143. I/O Device Status

As an alternative to the popup menus, you can obtain the same information by selecting the **Status** button in the various Edit Object dialogs or dialogs that show lists of IODF objects, such as the **Partitions** dialog or the **I/O Device List** dialog. If you are showing the status on the **Configuration diagram** (see the documentation on the **Refresh**, **Status** menu option, below), and you return to the diagram after having obtained status information for an individual object, you will notice that a + sign will appear to the right of the time stamp in the status bar at the bottom of the screen. This indicates that the status of one or more objects in the diagram is more recent than the indicated time of the last refresh.

In addition to the above, there are some special menu items for viewing your configuration or invoking dialogs that are of particular significance in connection with the I/O Operations tasks. Most of these are to be found under the main menu item **Operations** and are explained below.

Operations menu

The items of this top level menu item are enabled whenever you are logged on to HCD (i.e. not in stand-alone mode). It will appear to the left of the **Utilities** menu. The Operations feature of HCM consists of the following menu items:

1. **Refresh** This command is the primary means by which the data about the system status is obtained. This information is *volatile* i.e. it is not remembered across different HCM sessions, and if you change the **Focus on host**. The **Refresh** menu consists of the following four submenus:

   - **Status** This refreshes the status information for all objects in the configuration diagram. This allows you to see the changes you have made to the active sysplex by showing the updated objects in a different color. The status bar at the bottom of the screen will display the time at which this command was issued.

   - **Sysplex** This refreshes the current contents of the Sysplex without updating the status bar refresh timestamp. This command is issued implicitly whenever a configuration is opened, but the menu option is also provided so that the command can be manually issued to detect changes to the Sysplex or changes to the availability of I/O Operations that may have occurred during the session.

   - **Switch configurations** This refreshes the visualized matrix data for matrices that are being used in the current diagram. If one or more matrices are not available, messages will appear to tell the user that these matrices cannot be visualized. Visualization will proceed as if *none* had been selected as the Visualized Configuration for the affected switches. This action will automatically issue a "Refresh Sysplex" if one has not yet been issued. The status bar refresh timestamp is unaffected by this command.

   - **All** This refreshes all of the above information. The status bar refresh timestamp is updated by this command.

2. **Sense** This command is the only means by which "Sensed" data from the host is obtained. The menu item consists of three submenus listed below. Data obtained by the Sense operation appears in the appropriate columns of the **Edit/Locate** dialogs. This action will automatically issue a "Refresh Sysplex" if one has not yet been issued.

   - **Selected** This refreshes sensing information for the object(s) selected in the diagram.

   - **Visible** This refreshes information for all objects visible in the diagram.

   - **All** This refreshes sensing information for all objects in the configuration. Since this is likely to be a very long operation, a confirmation dialog will appear. The sensed information is stored in the HCM configuration file and is *non-volatile*. Hence, it is retained across HCM sessions. This information is also available in all List Boxes in which a **Columns** button is to be found e.g. the **Edit** and **Locate** dialogs.

   - **Clear Previous Sensing Data** This clears the sensed information for all objects in the configuration.

3. **Focus on host** This dialog allows you to select the host for which the status information is to be displayed.

4. **Next focus host** This is a shortcut to select (from the list that would appear in the Focus on host dialog) the next host as the focus host.

5. **Command...** This starts the I/O Operations command dialog which not only allows you to issue I/O Operations commands but also allows you to view the results of them (see "Issuing I/O Operations commands" on page 225).

6. **Preferences...** This dialog allows you to set various preferences specific only to the Operations feature of HCM e.g. the format of the system name is selected here.

Color dialog

This dialog allows you to assign the colors shown in the configuration diagram to the features they represent. It is called by the **View**, **Colors...** menu item. This dialog allows you to allocate your own colors for warnings and status.

If the user has selected the **Show status on diagram** checkbox in the **Colors** dialog, then the **Refresh**, **Status** menu option can be used to color objects to reflect status information with respect to the focus host. The following objects on the HCM diagram will be affected by the coloring:

- CHPIDs - Color will be determined by the status of the CHPID itself plus the SCP status of all CTC (or BCTC) devices that represent the CHPID in CTC connections to the focus host.
- Switches - Color will be determined by the operating status of the switch plus the SCP status of the switch's device.
- Integrated controllers - Color will be determined by the SCP status of the devices that the controller contains.
- Units within strings - Color will be determined by the SCP status of the devices that the unit contains.

Where more than one state or visual warning needs to be presented, the single most important state will be shown. Furthermore, for units and integrated controllers there will usually be more than one IODF object contained in the physical object. In such cases, the color displayed will reflect the status of the contained object with the most important state.

The status colors remain in effect on the diagram *as long as the view contains only objects that were visible at the time of the last status refresh*. The status time stamp in the status bar reflects this time. However, as soon as the view changes to include objects that were not visible when the time stamp was updated, then the coloring (and the time stamp) will be cleared.

This means that you can filter the diagram to show a large set of objects (e.g. all DASD in the active sysplex), invoke **Refresh**, **Status** once, and then switch back and forth between this larger view and views that show only subsets (e.g. use F4 to crop to one controller, and then F3 to go back to the larger view) without invalidating the status information. As soon as another object is shown in the view (e.g. a tape controller), then the colors and the status time stamp will be cleared, and "Refresh Status" would have to be issued again to show the status colors.

Manipulating and activating switch matrices

Switch matrices can reside inside an IODF or on a switch. HCM supports the manipulation of both kinds of matrices. Switch matrices that reside on switches can be listed, examined, modified, created or deleted in the same way as IODF matrices can. Every existing matrix is tagged with a label that indicates where it resides (in IODF or on a switch). When creating a matrix you can specify whether to place it in an IODF or on a switch.

The active switch configuration is treated in the same way as other switch configurations. It can be recognized by a special marker.

Certain restrictions apply to the manipulation of switch matrices. When access to the switch matrices is not possible, HCM informs the user accordingly.

HCM can activate any of the matrices that it can manipulate. This includes the active matrix. HCD will ensure that HCM can activate a matrix only when the active IODF is open.

Issuing I/O Operations commands

HCM offers a facility of passing I/O Operations commands to the HCD server. To issue I/O Operations commands, open the following **I/O Operations Command** window which can be found under the menu item **Operations, Command.**

Figure 144. I/O Operations Command window

In this window, you can enter commands that will be passed on to I/O Operations. The response of the executed command is then displayed in the window. You can scroll in the buffer to display the last few responses and save them to a file. To clear the buffer, click on the **Clear** button. It is also possible to retrieve the last few commands for executing again if necessary.

For valid I/O Operations commands see *System Automation for z/OS Operator's Commands*.

The return and reason codes resulting from the execution of the I/O Operations command are shown in the message CBDG133I. These codes are explained in *System Automation for z/OS Messages and Codes*.

Chapter 12. Reports and diagrams

HCM can produce a wide selection of reports and diagrams on a variety of printing devices, including plotters. Using the Windows print driver, HCM can work effectively with all printing devices supported by Windows. Since HCM is aware of the paper dimensions in the active printer, changing the paper size or orientation (portrait / landscape) will affect the appearance of the configuration diagram on the screen.

Use a printer resolution of 75 or 150 dpi (dots per inch): the graphics will print faster, although text is unaffected. Note that printing at 300 dpi or higher will require more printer memory. If this exceeds the capacity of your printer, print at a lower resolution.

HCM supports printing of hard copies in the same colors that appear on the screen.

Tip:

Some laser printers may print selected connections as patterned lines; you can generally force the printer to print solid lines by adjusting the printer driver settings.

Tip:

Before printing context diagrams, turn off the display of cabinets using the **View** menu. This will reduce the number of pages per context diagram.

Printing configuration diagrams

You can print configuration diagrams with the **Print Diagram...** command under the **File** menu. Configuration diagrams can be very large; you can reduce the number of pages needed by using legal sized paper (8 ¾″ x 14″) in landscape or portrait mode. You can print either the entire current diagram of the configuration diagram or a particular subset.

First ensure the current configuration diagram contains all the objects you want to print in the current view. Open the **Page Setup** dialog by choosing the **Page Setup** option in the **File** menu; select either Portrait or Landscape in the Orientation section. Clicking on the Printer button allows you to select the printer of your choice.

In the **Print Diagram** dialog, you can choose to print the exact portion of the diagram that appears on the screen, print the entire diagram on a single page, or scale the size of printed diagram.
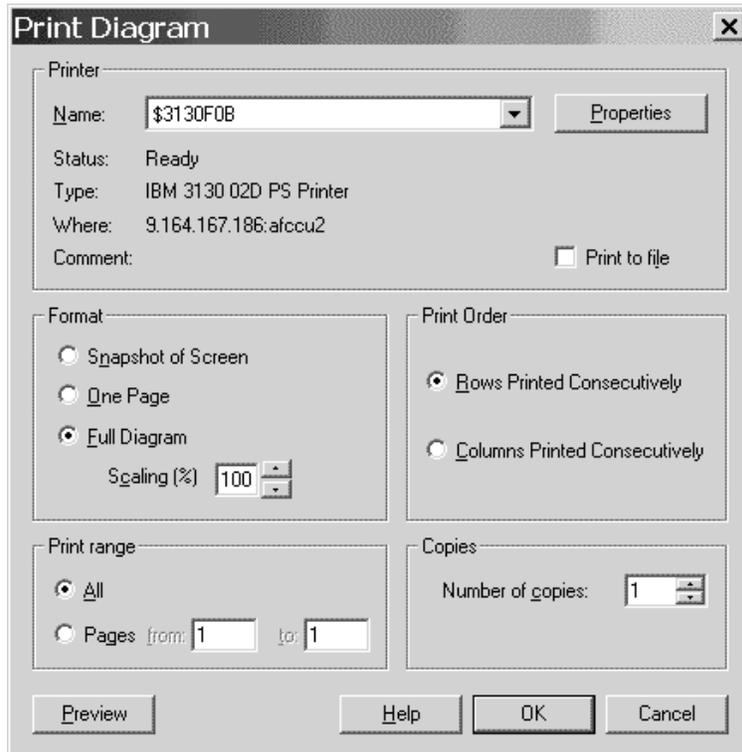
There are three options for printing the configuration diagram.

- **Snapshot of Screen**. Prints the exact portion of the configuration diagram that appears on the screen. Use this to produce diagrams of a particular subset of the configuration.

- **One Page Preview**. Prints the configuration diagram on one page, giving you an overview of the entire diagram. Note that configurations printed on one page will be difficult to read.

- **Full Diagram.** Prints the diagram over several pages, if necessary, at the scale specified in the Page Count section. You can select any scaling factor between 20% and 100%. Use the **Preview** button to determine how many pages will be required to print the diagram at the specified scale. Note that the **Show Page Breaks** option on the **View** menu only shows where page breaks will occur if the diagram is printed at 100%.

Printing reports and context diagrams

You can print reports and context diagrams using the **Print Reports** command under the **File** menu. A *context diagram* is a configuration diagram showing one particular object of interest and the equipment connected to it. Think of a context diagram as the configuration from the point of view of this object.

In the standalone mode you can only print reports of production configuration files. See "Printing reports, diagrams, cable labels" on page 92.

Open the **Printer Setup** dialog by choosing **Printer Setup...** from the **File** menu. Click the Setup option and select Portrait in the Orientation section.

Open the **Print Reports** dialog by choosing **Print Reports...** from the **File** menu. This dialog allows you to set parameters for the report you are printing. You can specify the report's title, two footnotes, and whether to include: cover page, table of contents, and index, as well as which sections will be in the body of the report and which objects will be reported on.
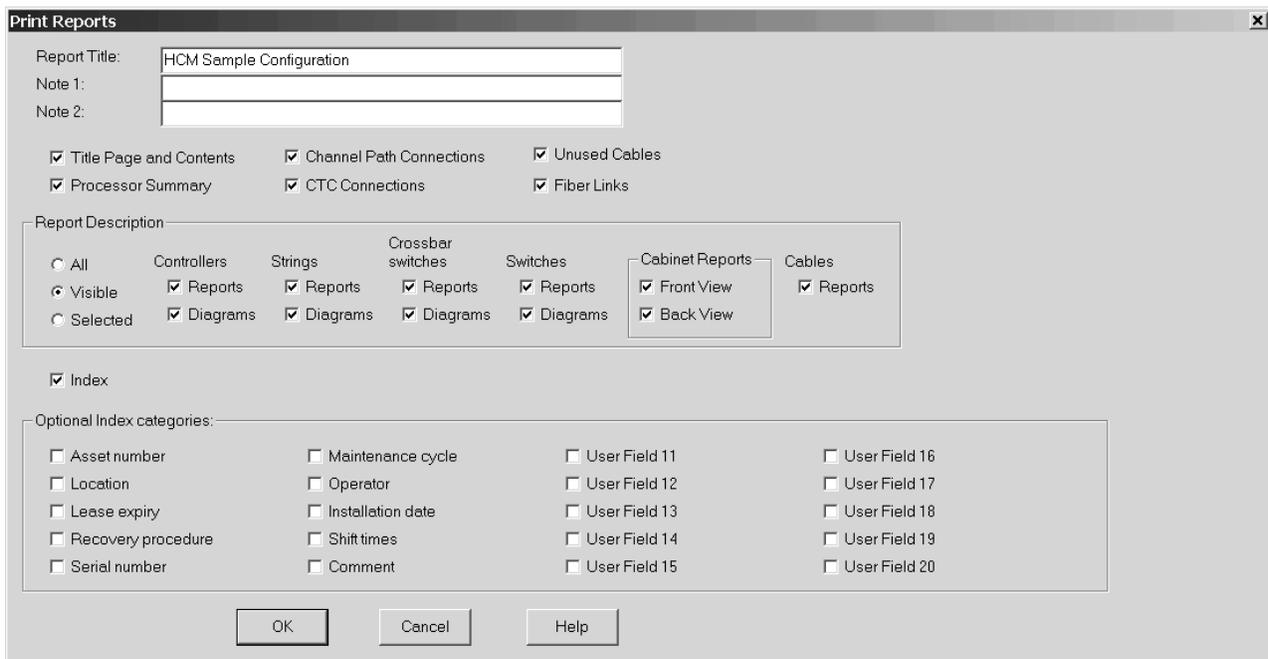
Figure 145. Print Reports Dialog

You can compose a report by including any or all of the following sections:

- **Title Page and Contents**. The title page contains the Report Title and Notes. The table of contents lists the sections that you include and the pages where each begins. Normally this option is included, but you may want to suppress these pages when printing reports on individual objects. To edit the title and footnotes that appear in the configuration diagrams and reports, open the **Titles and Footnotes** dialog of the **Edit** menu. Enter your values and press **OK**.

- **Processor Summary**. For each processor, prints a short summary that includes a description, and lists of partitions and CHPID types.

- **Channel Path Connections**. For each processor, prints a detailed CHPID report which describes the attached controllers and cable IDs.

- **CTC Connections.** Prints a detailed CTC report on all point-to-point CTCs and switched CTCs.
- **Fiber Links**. Prints a detailed report of all patchports and cables in each fiber link. The report is divided into sections for each type of ESCON or FICON object.
- **Controllers, Strings, Crossbar switches, ESCON or FICON switches, Cabinets, Cables** Specifies the types of objects for which detailed reports and context diagrams will be printed. You can choose to print reports and/or context diagrams for each object category selected. In addition, you can choose to print documentation for all objects in the object categories specified, only those visible (included) in the configuration diagram, or only those selected in the diagram.
- **Index.** Prints a detailed index for the report. The Optional Index categories section allows you to include user data fields as index items. For example, clicking on user data 1 which is, say, "Asset number" will result in all objects being indexed by asset number as long as they have entries in this user field.

Tip:

To print a context diagram of some or all CTC connections, choose **Filter diagram...** from the **View** menu and exclude all controller types except for CTC. You may also want to turn off the **Show All CHPIDs** option to ensure no extra objects are drawn simply because they are not yet connected to controllers. For indirect (via switches) CTCs, select the processors before printing. This will show the internal connectivity within the switches. Print the context diagram using the One Page Preview or Snapshot of Screen options in the **Print Diagram** dialog.

Tip:

If the Channel Path Connections option is selected, HCM will produce a full index because the entries in the index are determined by the objects referenced in a given report request.

Sample reports and context diagrams

The following pages contain one example of each type of report and context diagram that can be printed by selecting **Print Reports...** from the **File** menu.

Processor Summary

MCSSPR01 2084-C24 "XMP Processor for Documentation" [01]
    Mode: LPAR
    Support level: XMP, 3xx models, OSC

Channel subsystem MCSSPR01.0
Partitions (by image number)

1   LPARC01 1 CF/OS "CSS0/LPAR1 - 01"

2   LPARC02 2 CF/OS "CSS0/LPAR2 - 02"

3   LPARC03 3 OS "CSS0/LPAR3 - 03"

4   * 4 CF/OS "reserved LPAR for dynamic I/O"

5   LPARC05 5 OS "for null device candidate list"

CHPID	Type	# of Devices	MFS	Mode	PCHID	disable queue prioritization	Spanned in CSS 0	1	2	3	Partition Numbers 1	2	3	4	5
01	CNC	0	0	SHR							A	A	C		
02	CNC	0	0	SHR							A	A	C		
03	CNC	0	0	SHR							A	A	C		
04	IQD	0	16	SPAN			x	x			A		C		
05	IQD	0	16	SPAN			x	x			A				
06	IQD	0	16	SPAN			x	x			A				A
07	IQD	0	16	SPAN			x	x			A				A
08	OSC	0	0	SPAN			x	x			C	C	C		
10	OSC	0	0	SPAN			x	x			A	A			A
11	OSC	0	0	SPAN			x	x			A	A			A
12	CVC	9	0	REC							A				
13	CVC	0	0	DED								A			
14	CVC	0	0	DED							A				
16	FC	8	0	SPAN							A	A	A		
17	CNC	0	0	SHR							A	A	A		
18	FC	8	0	SPAN	31D		x	x			A				
1A	OSD	0	0	SPAN		Yes	x	x			A	C	A		
1B	OSD	0	0	SPAN		No	x	x			A	C	A		
1C	OSD	0	0	SPAN		No	x	x			A	C	A		
20	FC	24	0	SPAN			x	x			A	A	C		A
21	FC	8	0	SPAN			x	x			A	A	C		A
22	FC	8	0	SPAN			x	x			A	A	C		
23	FC	0	0	SHR			x	x			A	A	C		
24	FC	0	0	SHR			x	x			A	A	C		
25	FC	0	0	SHR			x	x			A	A	C		
26	FC	0	0	SHR			x	x			A	A	C		
27	FC	0	0	SHR			x	x			A	A	C		
28	FC	0	0	SHR			x	x			A	A	C		
29	FC	0	0	SHR			x	x			A	A	C		
2A	FC	0	0	SHR			x	x			A	A	C		
2B	FC	0	0	SHR			x	x			A	A	C		
2C	FC	0	0	SHR			x	x			A	A	C		
2D	FC	0	0	SHR			x	x			A	A	C		
2E	FC	8	0	SPAN			x	x			A	A	C		
2F	FC	8	0	SPAN			x	x			A	A	C		
30	FC	0	0	SHR			x	x			A	A	C		
31	FC	0	0	REC							A				
32	CTC	0	0	SHR									A		
33	CTC	0	0	DED									A		
34	CTC	0	0	DED									A		
35	CTC	0	0	DED									A		
36	CTC	0	0	DED									A		

Channel subsystem MCSSPR01.1
Partitions (by image number)
1   LPARC11 1 CF/OS "CSS1/LPAR1 - 11"

2   LPARC12 2 CF/OS "CSS1/LPAR2 - 12"

3   LPARC13 3 CF/OS "CSS1/LPAR3 - 13"

CHPID	Type	# of Devices	MFS	Mode	PCHID	disable queue prioritization	Spanned in CSS 0	1	2	3	Partition Numbers 1	2	3
04	IQD	0	16	SPAN			x	x			A		C
05	IQD	0	16	SPAN			x	x			A		
06	IQD	0	16	SPAN			x	x			A		
07	IQD	0	16	SPAN			x	x			A		
08	OSC	0	0	SPAN			x	x			A	A	A
09	OSC	0	0	SHR							A	A	A
10	OSC	0	0	SPAN			x	x			A	A	
11	OSC	0	0	SPAN			x	x			A	A	
12	CVC	0	0	DED									A
13	CVC	0	0	DED									A
14	CVC	0	0	DED									A
16	CNC	8	0	REC								C	A
17	CNC	8	0	REC								C	A
18	FC	0	0	SPAN	31D		x	x			A		
19	FC	0	0	SHR							C	A	A
1A	OSD	0	0	SPAN		Yes	x	x			C	A	C
1B	OSD	0	0	SPAN		No	x	x			C	A	C
1C	OSD	0	0	SPAN		No	x	x			C	A	C
20	FC	0	0	SPAN			x	x			A	A	C
21	FC	0	0	SPAN			x	x			A	A	C
22	FC	0	0	SPAN			x	x			A	A	C
23	FC	0	0	SPAN			x	x			A	A	C
24	FC	0	0	SPAN			x	x			A	A	C
25	FC	0	0	SPAN			x	x			A	A	C
26	FC	0	0	SPAN			x	x			A	A	C
27	FC	0	0	SPAN			x	x			A	A	C
28	FC	0	0	SPAN			x	x			A	A	C
29	FC	0	0	SPAN			x	x			A	A	C
2A	FC	0	0	SPAN			x	x			A	A	C
2B	FC	0	0	SPAN			x	x			A	A	C
2C	FC	0	0	SPAN			x	x			A	A	C
2D	FC	0	0	SPAN			x	x			A	A	C
2E	FC	0	0	SPAN			x	x			A	A	C
2F	FC	0	0	SPAN			x	x			A	A	C
30	FC	0	0	SPAN			x	x			A	A	C

Channel Path Connections

BOESYS 9672-R41 "OS/390 Development System"
    SN:0000009672

    Asset Mgmt .#:   8337539-77835
    Location: Main Complex

```
┌─────────────────────────────────────────────────────┐
│ 20 S CNC SHR                                          │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ F S in 3745-61A reaching String 00A0-00AC String 00AD ... │
  │  │                                                            │
  └──┴────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────┐
│ 21 S CNC SHR                                          │
│     # of Devices: 64                                  │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ SC0.A S in 3990-006-600 reaching RAMAC600                  │
  │  │         IBM 3990 Model 6 DASD Controller                   │
  └──┤         SN:13-93420                                        │
     └────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────┐
│ 22 S CNC SHR                                          │
│     # of Devices: 64                                  │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ SC1.A S in 3990-006-600 reaching RAMAC600                  │
  │  │         IBM 3990 Model 6 DASD Controller                   │
  └──┤         SN:13-93420                                        │
     └────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────┐
│ 23 S CNC SHR                                          │
│     # of Devices: 64                                  │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ SC0.B S in 3990-006-600 reaching RAMAC600                  │
  │  │         IBM 3990 Model 6 DASD Controller                   │
  └──┤         SN:13-93420                                        │
     └────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────┐
│ 24 S CNC SHR                                          │
│     # of Devices: 64                                  │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ SC1.B S in 3990-006-600 reaching RAMAC600                  │
  │  │         IBM 3990 Model 6 DASD Controller                   │
  └──┤         SN:13-93420                                        │
     └────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────┐
│ 25 S CNC SHR                                          │
│     # of Devices: 128                                 │
│     Access:        PROD, TEST                         │
└─┬───────────────────────────────────────────────────┘
  │
  mmf1080 85 MM Jumper
  │  ┌──────────────────────────────────────────────────────────┐
  │  │ SWCH_5.A9                                                  │
  └──┤         SN:1234567890                                      │
     └────────────────────────────────────────────────────────┘
```

BOESYS 9672-R41 "OS/390 Development System" continued...

26 S CNC SHR
    # of Devices: 128
    Access: PROD, TEST

mmf1078 85 MM Jumper
└─ SWCH_5.A2
├─   SN:1234567890

27 S CTC SHR
    # of Devices: 16
    Access: PROD, TEST

mmf1079 85 MM Jumper
└─ SWCH_5.A6
├─   SN:1234567890

28 S CNC SHR
    # of Devices: 144
    Access: PROD, TEST

mmf1081 85 MM Jumper
└─ SWCH_5.AA
├─   SN:1234567890

29 S CVC REC
    # of Devices: 8
    Access: PROD
    Candidate: (TEST)

└─ SWCH_5.AB
├─   SN:1234567890

2A P BL DED
    # of Devices: 2
    Access: PROD

└─ A P in 3274-41D-CE0 reaching CRTCE0
├─   SN:86588

└─ A P in 3274-41D-CC0 reaching CRTCC0
├─   SN:D2128

2B P BL REC
    # of Devices: 32
    Access: TEST
    Candidate: (PROD)

└─ SC0.B P in 3990-003-500 reaching String 500
            IBM 3990 Model 3 DASD Controller
├─   SN:32160

BOESYS 9672-R41 "OS/390 Development System" continued...

2C P BL REC
    Access: TEST
    Candidate: (PROD)

    A P in 6262 #2 Not connected

2D P BL DED
    Access: PROD

    C P in 3745-210 reaching String No Devices String 0918-091B

2E P BL DED
    # of Devices: 8
    Access: PROD

    SC1 C P in 3990-003-660 reaching String 660
          IBM 3990 Model 3 DASD Controller
          SN:33246

2F P BL REC
    Access: TEST
    Candidate: (PROD)

    B P in 3480-A22-480 reaching String 0480-0487 String 0488-048F
          IBM 3480 Tape Controller

30 P BL DED
    Access: PROD

    C P in 3480-A22-480 reaching String 0480-0487 String 0488-048F
          IBM 3480 Tape Controller

31 P BL DED
    # of Devices: 8
    Access: PROD

    SC0 B P in 3990-003-660 reaching String 660
          IBM 3990 Model 3 DASD Controller
          SN:33246

BOESYS 9672-R41 "OS/390 Development System" continued...

32 P BL DED
    # of Devices: 1
    Access: PROD

        A P in 3274-A00 reaching String 0A00

33 P BL DED
    # of Devices: 1
    Access: TEST

        A P in 3274-A10 reaching String 0A10

34 P BL DED
    # of Devices: 32
    Access: PROD

        SC0.C P in 3990-003-500 reaching String 500
             IBM 3990 Model 3 DASD Controller
             SN:32160

35 P BL REC
    # of Devices: 1
    Access: TEST
    Candidate: (PROD)

        X-Bar #1.2

        A P in 6262 #1 Not connected

        B P in 6262 #2 Not connected

36 P BL DED
    # of Devices: 32
    Access: PROD

        SC0.A P in 3990-003-500 reaching String 500
             IBM 3990 Model 3 DASD Controller
             SN:32160

37 P BL DED
    # of Devices: 8
    Access: PROD

        SC0.C P in 3990-003-660 reaching String 660
             IBM 3990 Model 3 DASD Controller
             SN:33246

Hardware Configuration Manager   2001-10-11 14:46 4

BOESYS 9672-R41 "OS/390 Development System" continued...

```
┌─────────────────────────────┐
│ 60 P BL DED                 │
│       Access:      PROD      │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────────────────┐
  ├───┤ B P in 3480-A22-490 reaching String 0490-0497                │
  │   │       IBM 3480 Tape Controller                               │
  ├   └─────────────────────────────────────────────────────────────┘

┌─────────────────────────────┐
│ 61 S CTC DED                │
│       # of Devices: 4       │
│       Access:      PROD      │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────┐
  ├───┤ BOESYS.66 S CNC DED                             │
  │   │       # of Devices: 4                           │
  ├   │       Access:      PROD                          │
  │   └─────────────────────────────────────────────────┘

┌─────────────────────────────┐
│ 62 P BL DED                 │
│       Access:      PROD      │
└─┬───────────────────────────┘
  ┴

┌─────────────────────────────┐
│ 63 P BY DED                 │
│       Access:      PROD      │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────────────────┐
  ├───┤ E P in 3745-61A reaching String 00A0-00AC String 00AD ...    │
  ├   └─────────────────────────────────────────────────────────────┘

┌─────────────────────────────┐
│ 64 P BL REC                 │
│       # of Devices: 32      │
│       Access:      TEST      │
│       Candidate:   (PROD)    │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────────────────┐
  ├───┤ SC1 D P in 3990-003-500 reaching String 500                  │
  │   │       IBM 3990 Model 3 DASD Controller                       │
  ├   │       SN:32160                                               │
  │   └─────────────────────────────────────────────────────────────┘

┌─────────────────────────────┐
│ 65 P BL REC                 │
│       # of Devices: 1       │
│       Access:      TEST      │
│       Candidate:   (PROD)    │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────────────────┐
  ├───┤ A P in 3274-41D-C60 reaching CRT C60                         │
  ├   │       SN:F1309                                               │
  │   └─────────────────────────────────────────────────────────────┘

┌─────────────────────────────┐
│ 66 S CNC DED                │
│       # of Devices: 4       │
│       Access:      PROD      │
└─┬───────────────────────────┘
  │   ┌─────────────────────────────────────────────────┐
  ├───┤ BOESYS.61 S CTC DED                             │
  │   │       # of Devices: 4                           │
  ├   │       Access:      PROD                          │
  │   └─────────────────────────────────────────────────┘
```

Channel Path Connections

80 S CNC SHR (Managed)
Managed: Yes I/O Cluster: UTCPLX02

SWCH_0.80

81 S CNC SHR (Managed)
Managed: Yes I/O Cluster: UTCPLX02

SWCH_0.81

82 S CNC SHR (Managed)
Managed: Yes I/O Cluster: UTCPLX02

SWCH_0.82

83 S CNC SHR (Managed)
Managed: Yes I/O Cluster: UTCPLX02

SWCH_0.83

90 S CNC SHR
# of Devices: 256
Access: XA, XB
Candidate: (XC)

10.A S in 2105 #1 reaching String 1000-1003 String 1004-10FF ...
        IBM Enterprise Storage Server (ESS)

91 S CNC SHR
Access: XA, XB, XC

SWCH_0.84

92 S CNC SHR (Managed)
Managed: Yes I/O Cluster: UTCPLX02

SWCH_0.85

ESERVER 2064-112 "eServer z900" continued...

93 S CNC SHR
        Access: XA, XB, XC

         SWCH_0.86

94 S FC DED "FC connecting to ESS"
        # of Devices: 252
        Access: XA

         SWCH_1.00

95 S FC SHR "FC connecting to ESS"
        # of Devices: 4
        Access: XA, XC

         11.B S in 2105 #1 reaching String 1000-1003 String 1004-10FF ...
            IBM Enterprise Storage Server (ESS)

96 S FC SHR "FC connecting to ESS"
        # of Devices: 4
        Access: XA, XC

         12.A S in 2105 #1 reaching String 1000-1003 String 1004-10FF ...
            IBM Enterprise Storage Server (ESS)

Channel Path Connections

Channel subsystem XMP2.0

```
0.10 S ICP SPAN
        Spanned with XMP2.0 XMP2.2
        # of Devices: 21
        Access:        PART01
```

```
        XMP2.1.57 S ICP SPAN
                Spanned with XMP2.1 XMP2.3
                # of Devices: 7
                Access:        OSPART
```

```
0.27 S CNC SHR PCHID=1C3 "ESCON channel"
        Access:        PART01
```

```
        SWCH_2.28
```

```
0.28 S CNC SHR PCHID=207 "ESCON channel"
        Access:        PART01
```

```
        SWCH_2.29
```

```
0.96 S ICP REC
        Access:        PART02
        Candidate:     (PART01)
```

Channel subsystem XMP2.1

```
1.56 S ICP SPAN
        Spanned with XMP2.1 XMP2.3
        # of Devices: 21
        Access:        OSPART
```

```
        XMP2.1.58 S ICP SPAN
                Spanned with XMP2.1 XMP2.3
                # of Devices: 21
                Access:        OSPART
```

```
1.57 S ICP SPAN
        Spanned with XMP2.1 XMP2.3
        # of Devices: 7
        Access:        OSPART
```

```
        XMP2.0.10 S ICP SPAN
                Spanned with XMP2.0 XMP2.2
                # of Devices: 21
                Access:        PART01
```

XMP2 2084-D32 "XMP2 Processor" continued...

1.58 S ICP SPAN
      Spanned with XMP2.1 XMP2.3
      # of Devices: 21
      Access: OSPART

    XMP2.1.56 S ICP SPAN
          Spanned with XMP2.1 XMP2.3
          # of Devices: 21
          Access: OSPART

1.59 S FC SHR PCHID=1C7
      Access: CFOSPART, OSPART
      Candidate: (OS2)

    SWCH_5.05

1.5A S FC SHR PCHID=039
      Access: OSPART
      Candidate: (CFOSPART, OS2)

    SWCH_5.08

1.5B S FC SHR PCHID=20D
      Access: OSPART
      Candidate: (OS2)

    SWCH_5.3B

Channel subsystem XMP2.2

2.10 S ICP SPAN
      Spanned with XMP2.0 XMP2.2
      # of Devices: 21
      Access: PART21, PART22

2.23 S FCP SHR PCHID=3D2
      # of Devices: 1
      Access: PART21

    A S in FCP #1 reaching String 4599

Channel subsystem XMP2.3

3.56 S ICP SPAN
Spanned with XMP2.1 XMP2.3
# of Devices: 21
Access: PARTCF, PARTOS
Candidate: (PARTCFOS)

3.57 S ICP SPAN
Spanned with XMP2.1 XMP2.3
# of Devices: 7
Access: NEWCFOS, PARTOS
Candidate: (PARTCFOS)

3.58 S ICP SPAN
Spanned with XMP2.1 XMP2.3
# of Devices: 21
Access: NEWCFOS, PARTOS
Candidate: (PARTCFOS)

3.78 O OSD SHR PCHID=19F
# of Devices: 1
Access: PARTCFOS
Candidate: (PARTOS)

A O in OSA #1 reaching String 8395

Channel-to-Channel Connections

CTC Connection

From CTC or FC CHPID:	BOESYS.27 S CTC SHR
Route:	ESCD_5(A6..AA)
From CNC/FCV or FC CHPID:	BOESYS.28 S CNC SHR

Details:

Image on BOESYS	Control Unit	Devices	OS	UA	Devices	OS	Control Unit	Image on BOESYS
TEST	0D02							TEST
TEST								PROD
PROD	0D02	0D00,8 (B)	Y	00	0D00,8 (B)	Y	0D03	TEST
PROD	0D02	0D08,8	Y	08	0D08,8	Y	0D03	TEST
PROD							0D03	PROD

CTC Connection

From CTC or FC CHPID:	BOESYS.61 S CTC DED
Route:	Point-to-Point
From CNC/FCV or FC CHPID:	BOESYS.66 S CNC DED

Details:

Image on BOESYS	Control Unit	Devices	OS	UA	Devices	OS	Control Unit	Image on BOESYS
PROD	1000	1000,4		00	1005,4		1005	PROD

CTC Connection

From CTC or FC CHPID:	ESERVER.A1 S FC SHR
Route:	ESCD_40(A1..B1)
From CNC/FCV or FC CHPID:	ESERVER2.B1 S FC SHR

Details:

Image on ESERVER	Control Unit	Devices	OS	UA	Devices	OS	Control Unit	Image on ESERVER2
XA	4200	4200,4		00	5200,4		5200	YA
XA	4210	4210,4		00	5200,4		5200	YB
XB	4200	4200,4		00	5210,4		5210	YA
XB	4210	4210,4		00	5210,4		5210	YB

CTC Connection

From CTC or FC CHPID:	ESERVER.A3 S FC SHR
Route:	Point-to-Point
From CNC/FCV or FC CHPID:	ESERVER2.B3 S FC SHR

Details:

Image on ESERVER	Control Unit	Devices	OS	UA	Devices	OS	Control Unit	Image on ESERVER2
XA	4100	4100,4		00	5100,4		5100	YA
XA	4110	4110,4		00	5100,4		5100	YB
XB	4100	4100,4		00	5110,4		5110	YA
XB	4110	4110,4		00	5110,4		5110	YB

Cabinet Cabinet #1 Low

Cabinet: Cabinet #1 Low
Model: Data Switch FMS
Short name: C1
Panel spaces: 1

LEGEND

Front View of Patchport

```
Connections
 on the back of
this patchport
_____

THIS PATCHPORT

        Connections
      on the front of
       this patchport
```

Back View of Patchport

```
Connections
 on the back of
this patchport

THIS PATCHPORT

        Connections
      on the front of
       this patchport
```
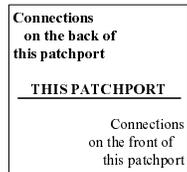
Uninstalled Patchport

```
THIS PATCHPORT
```

Cabinet C1, Panel 0, 6 by 6, Front View

Patchport Naming Format: "{ Data Switch FMS format }%CABINET%%PANEL/B%%cGROUP%%PORT/1%"

3990-003-240.SC0.A mmf1069 C20A1 trunk **== C10A1 ==** **mmf1024** **ESCD_5.80**	3990-003-240.SC1.A mmf1067 C20B1 Trunk **== C10B1 ==** **mmf1030** **ESCD_5.89**	3990-006-300.SC0.E mmf1057 C20C1 Trunk **== C10C1 ==** **mmf1036** **ESCD_5.83**	3990-003-360.SC1.E mmf5101 **== C10D1 ==** **mmf1046** **ESCD_5.98**	C20E1 Trunk **== C10E1 ==**	C20F1 Trunk **== C10F1 ==**
3990-003-240.SC0.B mmf1070 C20A2 Trunk **== C10A2 ==** **mmf1025** **ESCD_2.84**	3990-003-240.SC1.B mmf1068 C20B2 Trunk **== C10B2 ==** **mmf1031** **ESCD_2.8D**	3990-006-300.SC0.F mmf1058 C20C2 Trunk **== C10C2 ==** **mmf1037** **ESCD_2.87**	C20D2 Trunk **== C10D2 ==**	C20E2 Trunk **== C10E2 ==**	C20F2 Trunk **== C10F2 ==**
3990-003-200.SC1.A mmf1071 C20A3 Trunk **== C10A3 ==** **mmf1026** **ESCD_5.88**	3990-006-280.SC0.E mmf1061 C20B3 Trunk **== C10B3 ==** **mmf1032** **ESCD_5.82**	3990-006-300.SC1.E mmf1059 C20C3 Trunk **== C10C3 ==** **mmf1038** **ESCD_5.8B**	3990-006-380.SC0.E mmf1048 C20D3 Trunk **== C10D3 ==** **mmf1040** **ESCD_5.91**	C20E3 Trunk **== C10E3 ==**	C20F3 Trunk **== C10F3 ==**
3990-003-200.SC1.B mmf1072 C20A4 Trunk **== C10A4 ==** **mmf1027** **ESCD_2.8C**	3990-006-280.SC0.F mmf1062 C20B4 Trunk **== C10B4 ==** **mmf1033** **ESCD_2.86**	3990-006-300.SC1.F mmf1060 C20C4 Trunk **== C10C4 ==** **mmf1039** **ESCD_2.8F**	3990-006-380.SC0.F mmf1049 C20D4 Trunk **== C10D4 ==** **mmf1041** **ESCD_2.95**	**== C10E4 ==**	**== C10F4 ==**
3990-006-200.SC0.A mmf1065 C20A5 Trunk **== C10A5 ==** **mmf1028** **ESCD_5.81**	3990-006-280.SC1.E mmf1063 C20B5 Trunk **== C10B5 ==** **mmf1034** **ESCD_5.8A**	3990-003-360.SC0.E mmf5001 **== C10C5 ==** **mmf1044** **ESCD_5.90**	3990-006-380.SC1.E mmf1050 C20D5 Trunk **== C10D5 ==** **mmf1042** **ESCD_5.99**	**== C10E5 ==**	**== C10F5 ==**
3990-003-200.SC0.B mmf1066 C20A6 Trunk **== C10A6 ==** **mmf1029** **ESCD_2.85**	3990-006-280.SC1.F mmf1064 C20B6 Trunk **== C10B6 ==** **mmf1035** **ESCD_2.8E**	**== C10C6 ==**	3990-006-380.SC1.F mmf1051 C20D6 Trunk **== C10D6 ==** **mmf1043** **ESCD_2.9D**	**== C10E6 ==**	**== C10F6 ==**

Cabinet C1, Panel 0, 6 by 6, Back View

Patchport Naming Format: "{ Data Switch FMS format }%CABINET%%PANEL/B%%cGROUP%%PORT/1%"

C20F1 **Trunk** == **C10F1** ==	**C20E1** **Trunk** == **C10E1** ==	**3990-003-360.SC1.E** **mmf5101** == **C10D1** == mmf1046 ESCD_5.98	**3990-006-300.SC0.E** **mmf1057** **C20C1** **Trunk** == **C10C1** == mmf1036 ESCD_5.83	**3990-003-240.SC1.A** **mmf1067** **C20B1** **Trunk** == **C10B1** == mmf1030 ESCD_5.89	**3990-003-240.SC0.A** **mmf1069** **C20A1** **trunk** == **C10A1** == mmf1024 ESCD_5.80
C20F2 **Trunk** == **C10F2** ==	**C20E2** **Trunk** == **C10E2** ==	**C20D2** **Trunk** == **C10D2** ==	**3990-006-300.SC0.F** **mmf1058** **C20C2** **Trunk** == **C10C2** == mmf1037 ESCD_2.87	**3990-003-240.SC1.B** **mmf1068** **C20B2** **Trunk** == **C10B2** == mmf1031 ESCD_2.8D	**3990-003-240.SC0.B** **mmf1070** **C20A2** **Trunk** == **C10A2** == mmf1025 ESCD_2.84
C20F3 **Trunk** == **C10F3** ==	**C20E3** **Trunk** == **C10E3** ==	**3990-006-380.SC0.E** **mmf1048** **C20D3** **Trunk** == **C10D3** == mmf1040 ESCD_5.91	**3990-006-300.SC1.E** **mmf1059** **C20C3** **Trunk** == **C10C3** == mmf1038 ESCD_5.8B	**3990-006-280.SC0.E** **mmf1061** **C20B3** **Trunk** == **C10B3** == mmf1032 ESCD_5.82	**3990-003-200.SC1.A** **mmf1071** **C20A3** **Trunk** == **C10A3** == mmf1026 ESCD_5.88
== **C10F4** == *(crossed out)*	== **C10E4** == *(crossed out)*	**3990-006-380.SC0.F** **mmf1049** **C20D4** **Trunk** == **C10D4** == mmf1041 ESCD_2.95	**3990-006-300.SC1.F** **mmf1060** **C20C4** **Trunk** == **C10C4** == mmf1039 ESCD_2.8F	**3990-006-280.SC0.F** **mmf1062** **C20B4** **Trunk** == **C10B4** == mmf1033 ESCD_2.86	**3990-003-200.SC1.B** **mmf1072** **C20A4** **Trunk** == **C10A4** == mmf1027 ESCD_2.8C
== **C10F5** == *(crossed out)*	== **C10E5** == *(crossed out)*	**3990-006-380.SC1.E** **mmf1050** **C20D5** **Trunk** == **C10D5** == mmf1042 ESCD_5.99	**3990-003-360.SC0.E** **mmf5001** == **C10C5** == mmf1044 ESCD_5.90	**3990-006-280.SC1.E** **mmf1063** **C20B5** **Trunk** == **C10B5** == mmf1034 ESCD_5.8A	**3990-003-200.SC0.A** **mmf1065** **C20A5** **Trunk** == **C10A5** == mmf1028 ESCD_5.81
== **C10F6** == *(crossed out)*	== **C10E6** == *(crossed out)*	**3990-006-380.SC1.F** **mmf1051** **C20D6** **Trunk** == **C10D6** == mmf1043 ESCD_2.9D	== **C10C6** ==	**3990-006-280.SC1.F** **mmf1064** **C20B6** **Trunk** == **C10B6** == mmf1035 ESCD_2.8E	**3990-003-200.SC0.B** **mmf1066** **C20A6** **Trunk** == **C10A6** == mmf1029 ESCD_2.85

Fiber Link Report (CHPID Section)

Tx (dB)	Rx (dB)	Composition		
n/a	n/a	BOESYS.20	-jumper-	3745-61A.F
n/a	n/a	BOESYS.21	-jumper-	3990-006-600.SC0.A
n/a	n/a	BOESYS.22	-jumper-	3990-006-600.SC1.A
n/a	n/a	BOESYS.23	-jumper-	3990-006-600.SC0.B
n/a	n/a	BOESYS.24	-jumper-	3990-006-600.SC1.B
n/a	n/a	BOESYS.25	-mmf1080-	SWCH_5.A9
n/a	n/a	BOESYS.26	-mmf1078-	SWCH_5.A2
n/a	n/a	BOESYS.27	-mmf1079-	SWCH_5.A6
n/a	n/a	BOESYS.28	-mmf1081-	SWCH_5.AA
n/a	n/a	BOESYS.29	-jumper-	SWCH_5.AB
n/a	n/a	BOESYS.66	-jumper-	BOESYS.61
n/a	n/a	POKRUN.00	-jumper-	SWCH_2.80
n/a	n/a	POKRUN.01	-jumper-	SWCH_2.81
n/a	n/a	POKRUN.02	-jumper-	SWCH_2.82
n/a	n/a	POKRUN.03	-jumper-	SWCH_2.83
n/a	n/a	POKRUN.04	-jumper-	SWCH_2.88
n/a	n/a	POKRUN.05	-jumper-	SWCH_2.89
n/a	n/a	POKRUN.06	-jumper-	SWCH_2.8A
n/a	n/a	ESERVER.A0	-jumper-	SWCH_40.A0
n/a	n/a	ESERVER.A1	-jumper-	SWCH_40.A1
n/a	n/a	ESERVER.A2	-jumper-	SWCH_40.A2
n/a	n/a	ESERVER2.B0	-jumper-	SWCH_40.B0
n/a	n/a	ESERVER2.B1	-jumper-	SWCH_40.B1
n/a	n/a	ESERVER2.B2	-jumper-	SWCH_40.B2
n/a	n/a	ESERVER2.B3	-jumper-	ESERVER.A3

Fiber Link Report (Switch Section)

Tx (dB)	Rx (dB)	Composition						
n/a	n/a	SWCH_2.80	-jumper-	POKRUN.00				
n/a	n/a	SWCH_2.81	-jumper-	POKRUN.01				
n/a	n/a	SWCH_2.82	-jumper-	POKRUN.02				
n/a	n/a	SWCH_2.83	-jumper-	POKRUN.03				
n/a	n/a	SWCH_2.84	-mmfl025-	C10A2	-trunk-	C20A2	-mmfl070-	3990-003-240.SC0.B
n/a	n/a	SWCH_2.85	-mmfl029-	C10A6	-trunk-	C20A6	-mmfl066-	3990-003-200.SC0.B
n/a	n/a	SWCH_2.86	-mmfl033-	C10B4	-trunk-	C20B4	-mmfl062-	3990-006-280.SC0.F
n/a	n/a	SWCH_2.87	-mmfl037-	C10C2	-trunk-	C20C2	-mmfl058-	3990-006-300.SC0.F
n/a	n/a	SWCH_2.88	-jumper-	POKRUN.04				
n/a	n/a	SWCH_2.89	-jumper-	POKRUN.05				
n/a	n/a	SWCH_2.8A	-jumper-	POKRUN.06				
n/a	n/a	SWCH_2.8C	-mmfl027-	C10A4	-trunk-	C20A4	-mmfl072-	3990-003-200.SC1.B
n/a	n/a	SWCH_2.8D	-mmfl031-	C10B2	-trunk-	C20B2	-mmfl068-	3990-003-240.SC1.B
n/a	n/a	SWCH_2.8E	-mmfl035-	C10B6	-trunk-	C20B6	-mmfl064-	3990-006-280.SC1.F
n/a	n/a	SWCH_2.8F	-mmfl039-	C10C4	-trunk-	C20C4	-mmfl060-	3990-006-300.SC1.F
n/a	n/a	SWCH_2.94	-mmfl045-	C10C6	-jumper-	3990-003-360.SC0.F		
n/a	n/a	SWCH_2.95	-mmfl041-	C10D4	-trunk-	C20D4	-mmfl049-	3990-006-380.SC0.F
n/a	n/a	SWCH_2.9C	-jumper-	3990-003-360.SC1.F				
n/a	n/a	SWCH_2.9D	-mmfl043-	C10D6	-trunk-	C20D6	-mmfl051-	3990-006-380.SC1.F
n/a	n/a	SWCH_5.04	-jumper-	3990-006-380.SC0.G				
n/a	n/a	SWCH_5.05	-jumper-	3990-006-380.SC1.G				
n/a	n/a	SWCH_5.80	-mmfl024-	C10A1	-trunk-	C20A1	-mmfl069-	3990-003-240.SC0.A
n/a	n/a	SWCH_5.81	-mmfl028-	C10A5	-trunk-	C20A5	-mmfl065-	3990-003-200.SC0.A
n/a	n/a	SWCH_5.82	-mmfl032-	C10B3	-trunk-	C20B3	-mmfl061-	3990-006-280.SC0.E
n/a	n/a	SWCH_5.83	-mmfl036-	C10C1	-trunk-	C20C1	-mmfl057-	3990-006-300.SC0.E
n/a	n/a	SWCH_5.84	-jumper-	S->P				
n/a	n/a	SWCH_5.88	-mmfl026-	C10A3	-trunk-	C20A3	-mmfl071-	3990-003-200.SC1.A
n/a	n/a	SWCH_5.89	-mmfl030-	C10B1	-trunk-	C20B1	-mmfl067-	3990-003-240.SC1.A
n/a	n/a	SWCH_5.8A	-mmfl034-	C10B5	-trunk-	C20B5	-mmfl063-	3990-006-280.SC1.E
n/a	n/a	SWCH_5.8B	-mmfl038-	C10C3	-trunk-	C20C3	-mmfl059-	3990-006-300.SC1.E
n/a	n/a	SWCH_5.90	-mmfl044-	C10C5	-trunk-	C20C5	-mmfl052-	3990-003-360.SC0.E
n/a	n/a	SWCH_5.91	-mmfl040-	C10D3	-trunk-	C20D3	-mmfl048-	3990-006-380.SC0.E
n/a	n/a	SWCH_5.98	-mmfl046-	C10D1	-jumper-	3990-003-360.SC1.E		
n/a	n/a	SWCH_5.99	-mmfl042-	C10D5	-trunk-	C20D5	-mmfl050-	3990-006-380.SC1.E
n/a	n/a	SWCH_5.A2	-mmfl078-	BOESYS.26				
n/a	n/a	SWCH_5.A6	-mmfl079-	BOESYS.27				
n/a	n/a	SWCH_5.A9	-mmfl080-	BOESYS.25				
n/a	n/a	SWCH_5.AA	-mmfl081-	BOESYS.28				
n/a	n/a	SWCH_5.AB	-jumper-	BOESYS.29				
n/a	n/a	SWCH_40.A0	-jumper-	ESERVER.A0				
n/a	n/a	SWCH_40.A1	-jumper-	ESERVER.A1				
n/a	n/a	SWCH_40.A2	-jumper-	ESERVER.A2				
n/a	n/a	SWCH_40.B0	-jumper-	ESERVER2.B0				
n/a	n/a	SWCH_40.B1	-jumper-	ESERVER2.B1				
n/a	n/a	SWCH_40.B2	-jumper-	ESERVER2.B2				

Fiber Link Report (Controller Section)

Tx (dB)	Rx (dB)	Composition
n/a	n/a	3745-61A.F -jumper- BOESYS.20
n/a	n/a	3990-003-200.SC0.A -mmf1065- C20A5 -trunk- C10A5 -mmf1028- SWCH_5.81
n/a	n/a	3990-003-200.SC0.B -mmf1066- C20A6 -trunk- C10A6 -mmf1029- SWCH_2.85
n/a	n/a	3990-003-200.SC1.A -mmf1071- C20A3 -trunk- C10A3 -mmf1026- SWCH_5.88
n/a	n/a	3990-003-200.SC1.B -mmf1072- C20A4 -trunk- C10A4 -mmf1027- SWCH_2.8C
n/a	n/a	3990-003-240.SC0.A -mmf1069- C20A1 -trunk- C10A1 -mmf1024- SWCH_5.80
n/a	n/a	3990-003-240.SC0.B -mmf1070- C20A2 -trunk- C10A2 -mmf1025- SWCH_2.84
n/a	n/a	3990-003-240.SC1.A -mmf1067- C20B1 -trunk- C10B1 -mmf1030- SWCH_5.89
n/a	n/a	3990-003-240.SC1.B -mmf1068- C20B2 -trunk- C10B2 -mmf1031- SWCH_2.8D
n/a	n/a	3990-003-360.SC0.E -mmf1052- C20C5 -trunk- C10C5 -mmf1044- SWCH_5.90
n/a	n/a	3990-003-360.SC0.F -jumper- C10C6 -mmf1045- SWCH_2.94
n/a	n/a	3990-003-360.SC1.E -jumper- C10D1 -mmf1046- SWCH_5.98
n/a	n/a	3990-003-360.SC1.F -jumper- SWCH_2.9C
n/a	n/a	3990-006-280.SC0.E -mmf1061- C20B3 -trunk- C10B3 -mmf1032- SWCH_5.82
n/a	n/a	3990-006-280.SC0.F -mmf1062- C20B4 -trunk- C10B4 -mmf1033- SWCH_2.86
n/a	n/a	3990-006-280.SC1.E -mmf1063- C20B5 -trunk- C10B5 -mmf1034- SWCH_5.8A
n/a	n/a	3990-006-280.SC1.F -mmf1064- C20B6 -trunk- C10B6 -mmf1035- SWCH_2.8E
n/a	n/a	3990-006-300.SC0.E -mmf1057- C20C1 -trunk- C10C1 -mmf1036- SWCH_5.83
n/a	n/a	3990-006-300.SC0.F -mmf1058- C20C2 -trunk- C10C2 -mmf1037- SWCH_2.87
n/a	n/a	3990-006-300.SC1.E -mmf1059- C20C3 -trunk- C10C3 -mmf1038- SWCH_5.8B
n/a	n/a	3990-006-300.SC1.F -mmf1060- C20C4 -trunk- C10C4 -mmf1039- SWCH_2.8F
n/a	n/a	3990-006-380.SC0.E -mmf1048- C20D3 -trunk- C10D3 -mmf1040- SWCH_5.91
n/a	n/a	3990-006-380.SC0.F -mmf1049- C20D4 -trunk- C10D4 -mmf1041- SWCH_2.95
n/a	n/a	3990-006-380.SC0.G -jumper- SWCH_5.04
n/a	n/a	3990-006-380.SC1.E -mmf1050- C20D5 -trunk- C10D5 -mmf1042- SWCH_5.99
n/a	n/a	3990-006-380.SC1.F -mmf1051- C20D6 -trunk- C10D6 -mmf1043- SWCH_2.9D
n/a	n/a	3990-006-380.SC1.G -jumper- SWCH_5.05
n/a	n/a	3990-006-600.SC0.A -jumper- BOESYS.21
n/a	n/a	3990-006-600.SC0.B -jumper- BOESYS.23
n/a	n/a	3990-006-600.SC1.A -jumper- BOESYS.22
n/a	n/a	3990-006-600.SC1.B -jumper- BOESYS.24

Fiber Link Report (Converter Section)

Tx (dB)	Rx (dB)	Composition
n/a	n/a	S->P -jumper- SWCH_5.84

3990-003-360 SN:35563

SN:35563

Asset Mgmt .#:   8337539-77835
Location: Main Complex
Usage: Databases

3990-003-360.SC0
    Control Units:
        0340   **3990-3**   **SN:35563**
                Devices: 340,64
                BOESYS: CHPIDs: 25,28
                POKRUN: CHPIDs: 00

    Channel Interfaces:

SC0.A P	No connection				
SC0.B P	No connection				
SC0.C P	No connection				
SC0.D P	No connection				
SC0.E S	**ESCD_5.90**				
	In:	via C10C5	mmf5001	75	MM Jumper
		on ESCD_5.90	mmf1044	60	MM Jumper
SC0.F S	**ESCD_2.94**				
	In:	**on ESCD_2.94**	mmf5002	75	MM Jumper

    Device Interfaces:

SC0.SP0	**A in String 0360-037F**	mmf4001	50	Blue Full
	A in String 340-35F			
SC0.SP1	**B in String 0360-037F**	mmf4002	50	Blue Full
	B in String 340-35F			

3990-003-360.SC1
    Control Units:
        0341   **3990-3**   **SN:35563**
                Devices: 340,64
                BOESYS: CHPIDs: 26
                POKRUN: CHPIDs: 01

    Channel Interfaces:

SC1.A P	No connection				
SC1.B P	No connection				
SC1.C P	No connection				
SC1.D P	No connection				
SC1.E S	**ESCD_5.98**				
	In:	via C10D1	mmf5101	90	MM Jumper
		on ESCD_5.98	mmf1046	60	MM Jumper
SC1.F S	**ESCD_2.9C**				
	In:	**on ESCD_2.9C**	mmf5102	90	MM Jumper

    Device Interfaces:

SC1.SP2	**C in String 0360-037F**	mmf4003	50	Blue Full
	C in String 340-35F			
SC1.SP3	**D in String 0360-037F**	mmf4004	50	Blue Full
	D in String 340-35F			

3990-003-360
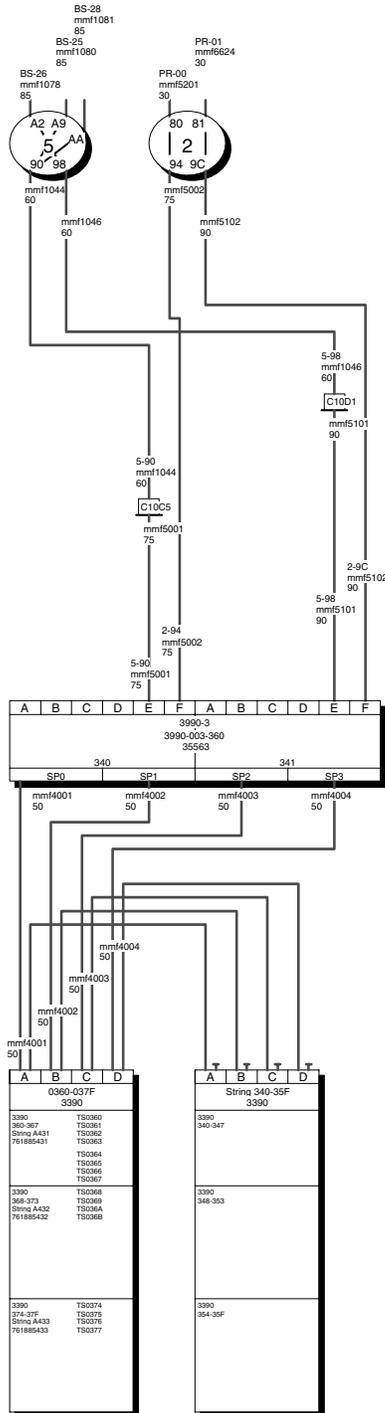
String 0360-037F SN:761885431

SN:761885431

Asset Mgmt .#:   8337539-77835
Location: Main complex
Usage: Databases

Devices:
BOESYS, POKRUN
    Control Units: 0340   0341
    360,32   3390   SN:761885431   "Dev.Nr. 0360"

String Interfaces:

A	In:	3990-003-360.SC0.SP0		mmf4001	50	Blue Full
	Out:	String 340-35F.A				
B	In:	3990-003-360.SC0.SP1		mmf4002	50	Blue Full
	Out:	String 340-35F.B				
C	In:	3990-003-360.SC1.SP2		mmf4003	50	Blue Full
	Out:	String 340-35F.C				
D	In:	3990-003-360.SC1.SP3		mmf4004	50	Blue Full
	Out:	String 340-35F.D				

Units:
String A431
        SN:761885431
    Asset Mgmt .#:   8337539-77835
    Location: Main Complex
    Usage: Databases

360	TS0360	361	TS0361	362	TS0362	363	TS0363	364	TS0364
365	TS0365	366	TS0366	367	TS0367				

String A432
        SN:761885432
    Asset Mgmt .#:   8337539-77835
    Location: Main Complex
    Usage: Databases

368	TS0368	369	TS0369	36A	TS036A	36B	TS036B	36C,8

String A433
        SN:761885433
    Asset Mgmt .#:   8337539-77835
    Location: Main Complex
    Usage: Databases

374	TS0374	375	TS0375	376	TS0376	377	TS0377	378,8

String 0360-037F
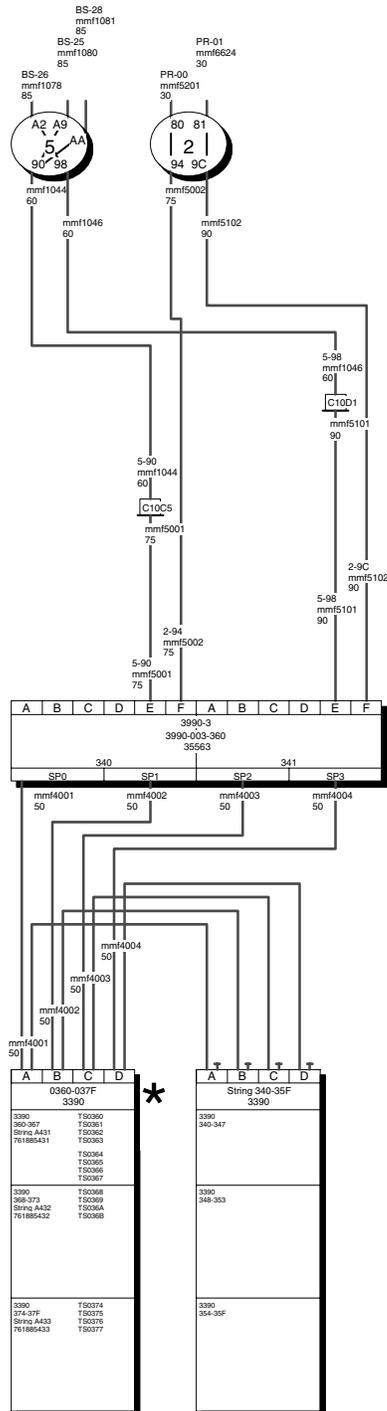
Crossbar #1 [X1]

SN:7743298543543254

Asset Mgmt .#:   8337539-77835
Location: Main complex

Normal / Default configuration:
    Weekday

Configurations:
    Weekday
    Weekend

Channel interfaces:

1	**POKRUN.09**				
	In:	on POKRUN.09	mmf9945	30	Blue Full
	Out:	Terminated			
2	**BOESYS.35**				
	In:	on BOESYS.35	mmf9946	30	Blue Full
	Out:	to 6262 #1.A	mmf9947	150	Blue Full
3	**POKRUN.0A**				
	In:	on POKRUN.0A	mmf9948	40	Blue Full
	Out:	Terminated			
4	No connection				

Control Unit Interfaces:

A1	**A in 6262 #2 Not connected**	mmf9950	75	Blue Full
B2	**A in 6262-022 reaching String 000F**	mmf9954	75	Blue Full
C3	No connection			
D4	No connection			

Crossbar #1 configuration: Weekday

```
P       B       P
O       O       O
K       O       K
R       E       R
U       S       U
N       Y       N
.       S       .
0       .       0
9       3       A
        5
```

```
┌───┬───┬───┬───┐
│ 1 │ 2 │ 3 │ 4 │
├───┼───┼───┼───┤
│   │ ● │   │   │  A1    6262 #2.A
│   │   │ ● │   │  B2    6262-022.A
│ ● │   │   │   │  C3
│ ● │   │   │   │  D4
└───┴───┴───┴───┘
```

Channel Interfaces in use:

1	POKRUN.09	C3	No connection
		D4	No connection
2	BOESYS.35	A1	A in 6262 #2 Not connected
3	POKRUN.0A	B2	A in 6262-022 reaching String 000F

Control Unit Interfaces in use:

A1 A in 6262 #2 Not connected
        2   BOESYS.35

B2 A in 6262-022 reaching String 000F
        3   POKRUN.0A

Crossbar #1 configuration: Weekend

```
P       B       P
O       O       O
K       E       K
R       S       R
U       Y       U
N       S       N
.       .       .
0       3       0
9       5       A

    ┌───┬───┬───┬───┐
    │ 1 │ 2 │ 3 │ 4 │
    ├───┴───┴───┴───┤────┐
    │ ●             │ A1 │   6262 #2.A
    │               │────┤
    │     ●         │ B2 │   6262-022.A
    │               │────┤
    │ ●             │ C3 │
    │               │────┤
    │     ●         │ D4 │
    └───────────────┴────┘
```
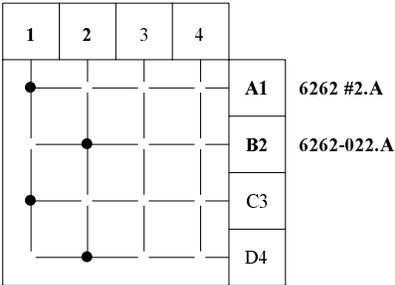
Channel Interfaces in use:

| 1 | **POKRUN.09** | **A1** | A in 6262 #2 Not connected |
| | | **C3** | No connection |

| 2 | **BOESYS.35** | **B2** | A in 6262-022 reaching String 000F |
| | | **D4** | No connection |

Control Unit Interfaces in use:

A1   **A in 6262 #2 Not connected**
       1   POKRUN.09

B2   **A in 6262-022 reaching String 000F**
       2   BOESYS.35

Crossbar #1 (1 of 2)

BOESYS 9672-R41

OS/390 Development System

PROD

TEST

| 35 | 61 | 66 | 6B |

mmf9946
30

PR-0A
mmf9948
40

BS-35
mmf9946
30

PR-09
mmf9945
30

| 1 | 2 | 3 | 4 |

| A1 |
| B2 |
| C3 |
| D4 |

Crossbar #1

mmf9950(75)
mmf9954(75)

BS-35

X1-B2
mmf9954
75

X1-A1
mmf9950
75

A
4248
6262-022
00F

A	B
6262	
6262 #2	
2627	

Crossbar #1 (2 of 2)

```
                              POKRUN 2003-103
                              OS/390 - Production System
     09 0A 10
     mmf9945
     30
          mmf9948
          40
```
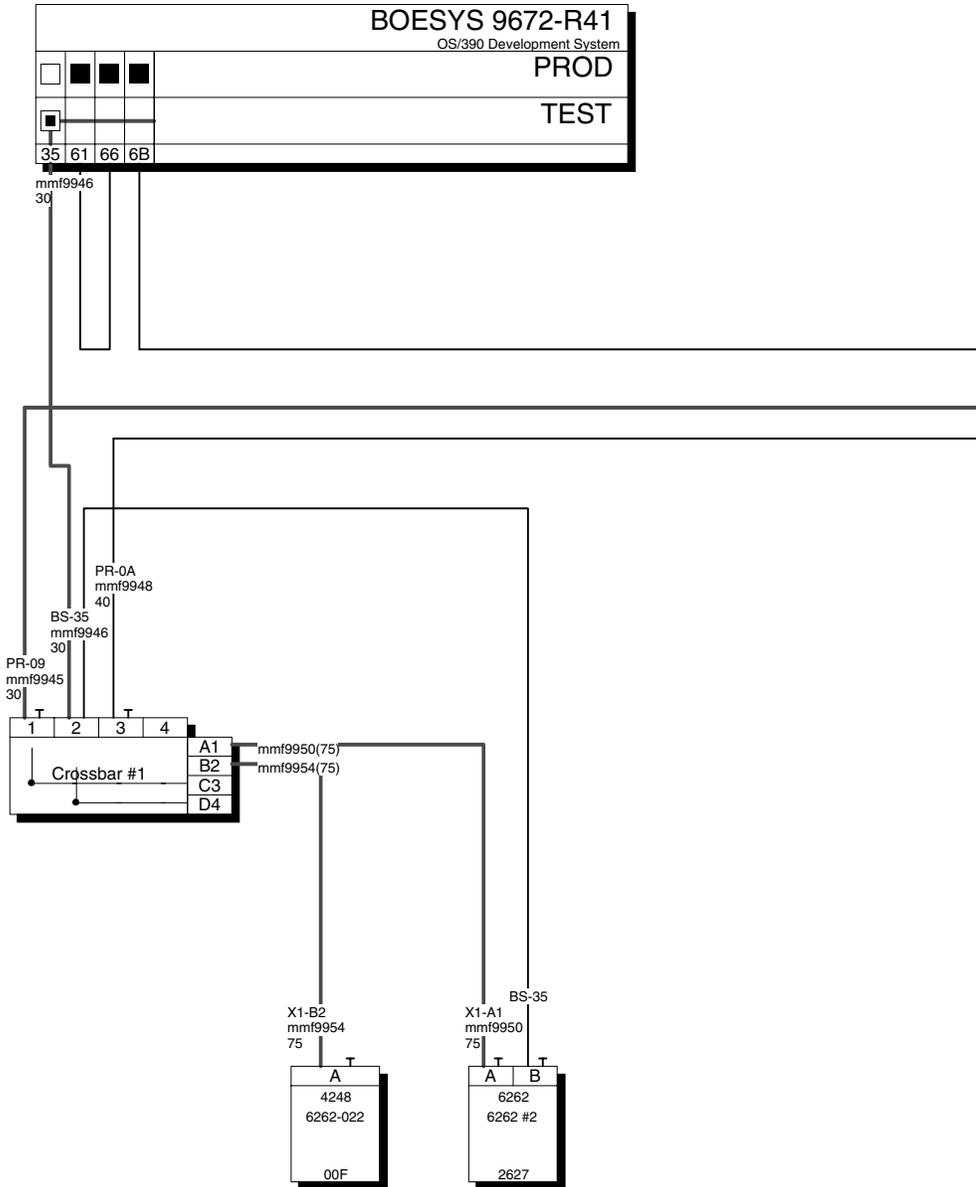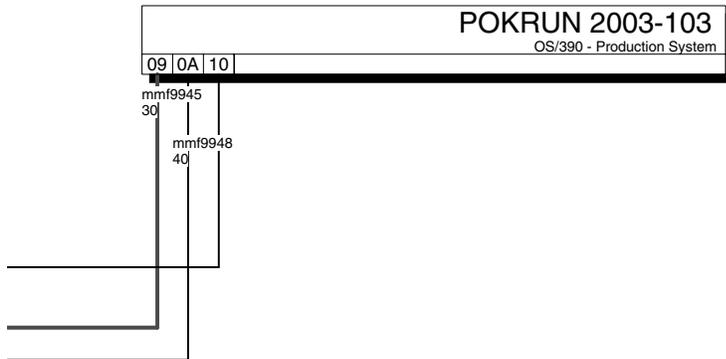
ESCD_5

Model: 9032-5
Used for FC-Bridge card
SN:8430954328
Asset Mgmt .#:   8337539-77835

Control Unit:
     5442   **9032-5**   **SN:8430954328**   **"Used for FC-Bridge card"**
           Devices: 5442,1
           BOESYS: CHPIDs: 25

Device:
    BOESYS
           Control Units: 5442
           5442,1   9032-5   SN:8430954328   "Used for FC-Bridge card"

PORTS ON THE PROCESSOR SIDE:

A2 LED	**on BOESYS.26**	mmfl078	85	MM Jumper
A6 LED	**on BOESYS.27**	mmfl079	85	MM Jumper
A9 LED	**on BOESYS.25**	mmfl080	85	MM Jumper
AA LED	**on BOESYS.28**	mmfl081	85	MM Jumper
AB LED	**on BOESYS.29**			

PORTS ON THE CONTROL UNIT SIDE:

04 LED   **to SC0.G in 3990-006-380**
                reaching String 0380-039F String-3A0

05 LED   **to SC1.G in 3990-006-380**
                reaching String 0380-039F String-3A0

80 LED	via C10A1	mmfl024	60	MM Jumper
	via C20A1	trunk	80	MM Jumper
	to SC0.A in 3990-003-240 reaching String 0200-021F String 0220-02	mmfl069	100	MM Jumper
81 LED	via C10A5	mmfl028	60	MM Jumper
	via C20A5			
	to SC0.A in 3990-003-200 reaching String 0240-025F String 0260-02	mmfl065	100	MM Jumper
82 LED	via C10B3	mmfl032	60	MM Jumper
	via C20B3			
	to SC0.E in 3990-006-280 reaching RAMAC280	mmfl061	100	MM Jumper
83 LED	via C10C1	mmfl036	60	MM Jumper
	via C20C1			
	to SC0.E in 3990-006-300 reaching RAMAC300	mmfl057	100	MM Jumper
84 LED	via S->P			
	to A in 3803-460 reaching String 0460-0467			
88 LED	via C10A3	mmfl026	60	MM Jumper
	via C20A3			
	to SC1.A in 3990-003-200 reaching String 0240-025F String 0260-02	mmfl071	100	MM Jumper
89 LED	via C10B1	mmfl030	60	MM Jumper
	via C20B1			
	to SC1.A in 3990-003-240 reaching String 0200-021F String 0220-02	mmfl067	100	MM Jumper

8A LED	via C10B5		mmfl034	60	MM Jumper
	via C20B5				
	to SC1.E in 3990-006-280 reaching RAMAC280		mmfl063	100	MM Jumper
8B LED	via C10C3		mmfl038	60	MM Jumper
	via C20C3				
	to SC1.E in 3990-006-300 reaching RAMAC300		mmfl059	100	MM Jumper
90 LED	via C10C5		mmfl044	60	MM Jumper
	to SC0.E in 3990-003-360 reaching String 0360-037F String 340-35F		mmf5001	75	MM Jumper
91 LED	via C10D3		mmfl040	60	MM Jumper
	via C20D3				
	to SC0.E in 3990-006-380 reaching String 0380-039F String-3A0		mmfl048	100	MM Jumper
98 LED	via C10D1		mmfl046	60	MM Jumper
	to SC1.E in 3990-003-360 reaching String 0360-037F String 340-35F		mmf5101	90	MM Jumper
99 LED	via C10D5		mmfl042	60	MM Jumper
	via C20D5				
	to SC1.E in 3990-006-380 reaching String 0380-039F String-3A0		mmfl050	100	MM Jumper
FE (internal)	**to A in 9032 #5**				
	reaching String 5442				

Hardware Configuration Manager
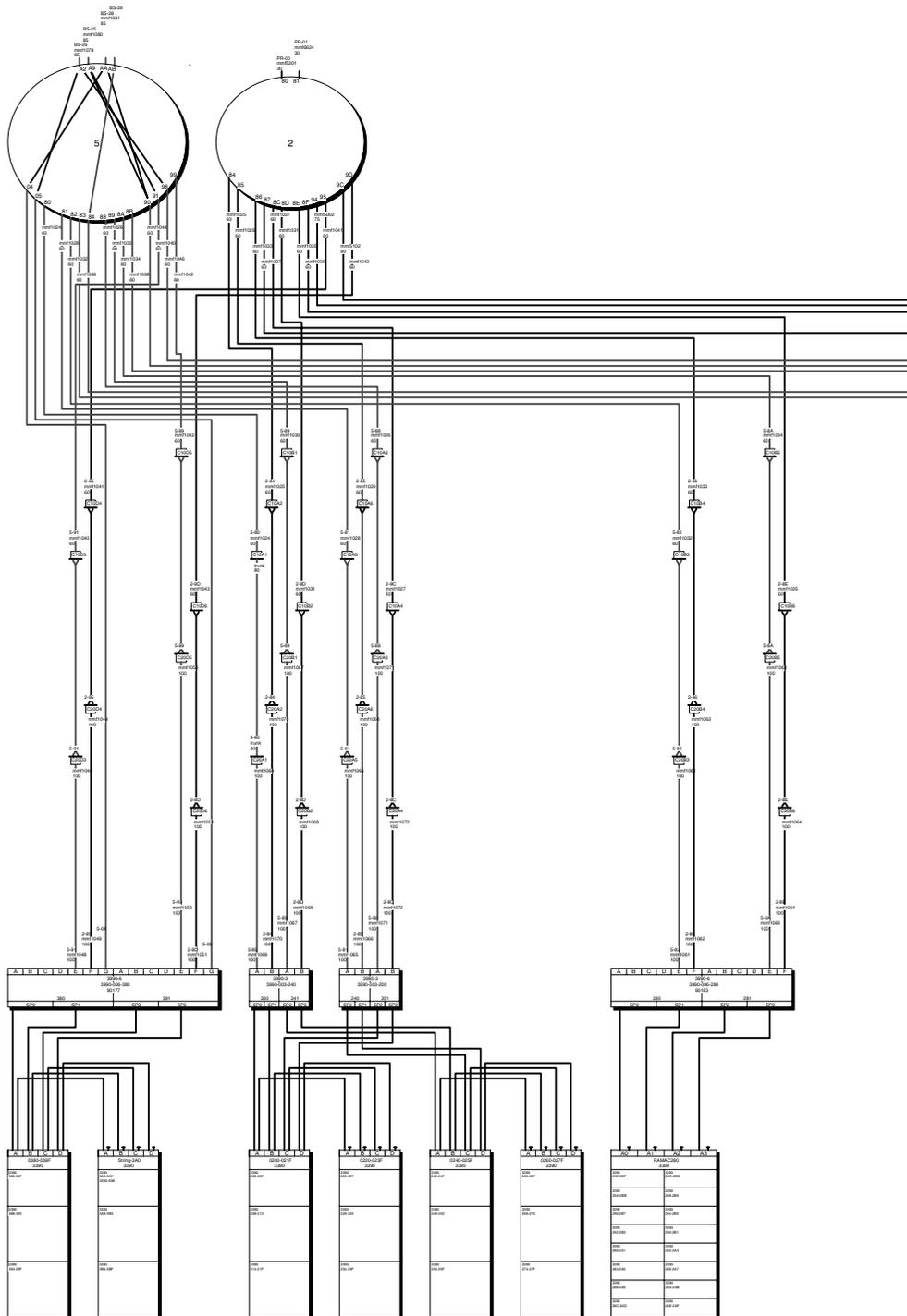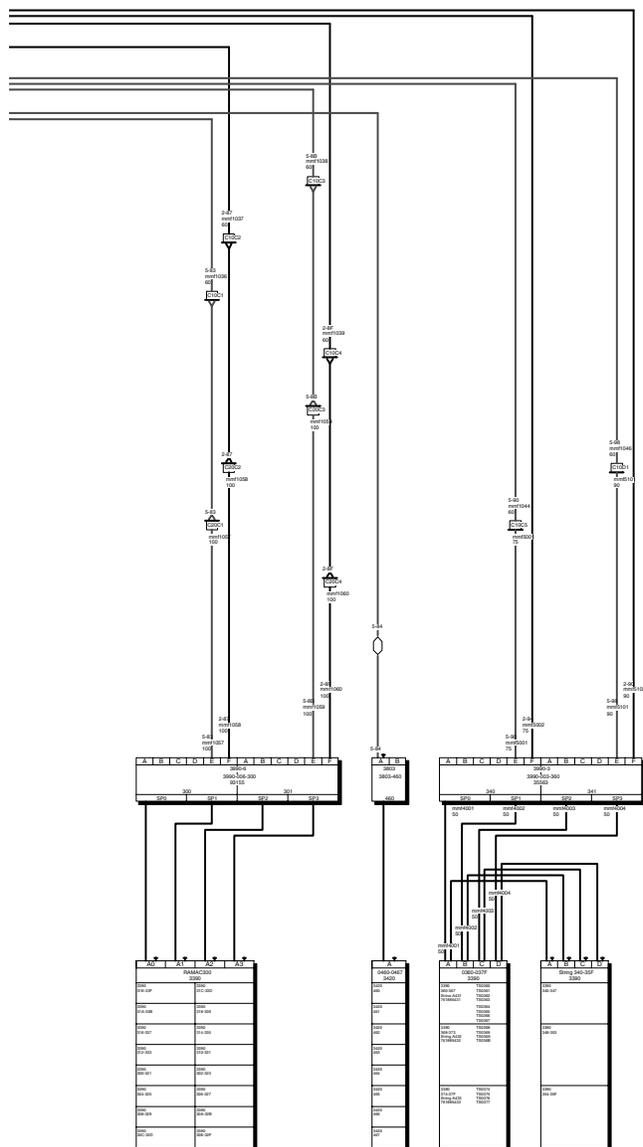
Cables

From	ID	Len	Type	Page #	To	db Loss A & B	
ESCD_5.80	mmfl024	60	MM Jumper		C10A1	n/a	n/a
ESCD_5.88	mmfl026	60	MM Jumper		C10A3	n/a	n/a
ESCD_5.81	mmfl028	60	MM Jumper		C10A5	n/a	n/a
ESCD_5.89	mmfl030	60	MM Jumper		C10B1	n/a	n/a
ESCD_5.82	mmfl032	60	MM Jumper		C10B3	n/a	n/a
ESCD_5.8A	mmfl034	60	MM Jumper		C10B5	n/a	n/a
ESCD_5.83	mmfl036	60	MM Jumper		C10C1	n/a	n/a
ESCD_5.8B	mmfl038	60	MM Jumper		C10C3	n/a	n/a
ESCD_5.91	mmfl040	60	MM Jumper		C10D3	n/a	n/a
ESCD_5.99	mmfl042	60	MM Jumper		C10D5	n/a	n/a
ESCD_5.90	mmfl044	60	MM Jumper		C10C5	n/a	n/a
ESCD_5.98	mmfl046	60	MM Jumper		C10D1	n/a	n/a
C20D3	mmfl048	100	MM Jumper		3990-006-380.SC0.E	n/a	n/a
C20D5	mmfl050	100	MM Jumper		3990-006-380.SC1.E	n/a	n/a
C20C1	mmfl057	100	MM Jumper		3990-006-300.SC0.E	n/a	n/a
C20C3	mmfl059	100	MM Jumper		3990-006-300.SC1.E	n/a	n/a
C20B3	mmfl061	100	MM Jumper		3990-006-280.SC0.E	n/a	n/a
C20B5	mmfl063	100	MM Jumper		3990-006-280.SC1.E	n/a	n/a
C20A5	mmfl065	100	MM Jumper		3990-003-200.SC0.A	n/a	n/a
C20B1	mmfl067	100	MM Jumper		3990-003-240.SC1.A	n/a	n/a
C20A1	mmfl069	100	MM Jumper		3990-003-240.SC0.A	n/a	n/a
C20A3	mmfl071	100	MM Jumper		3990-003-200.SC1.A	n/a	n/a
BOESYS.26	mmfl078	85	MM Jumper		ESCD_5.A2	n/a	n/a
BOESYS.27	mmfl079	85	MM Jumper		ESCD_5.A6	n/a	n/a
BOESYS.25	mmfl080	85	MM Jumper		ESCD_5.A9	n/a	n/a
BOESYS.28	mmfl081	85	MM Jumper		ESCD_5.AA	n/a	n/a
3990-003-360.SC0.SP0	mmf4001	50	Blue Full		String 0360-037F.A		
3990-003-360.SC0.SP1	mmf4002	50	Blue Full		String 0360-037F.B		
3990-003-360.SC1.SP2	mmf4003	50	Blue Full		String 0360-037F.C		
3990-003-360.SC1.SP3	mmf4004	50	Blue Full		String 0360-037F.D		
C10C5	mmf5001	75	MM Jumper		3990-003-360.SC0.E	n/a	n/a
C10D1	mmf5101	90	MM Jumper		3990-003-360.SC1.E	n/a	n/a
C10A1	trunk	80	MM Jumper		C20A1	n/a	n/a

Sample cable labels

The following is a sample of cable labels as printed on a printable label form.

EMIFSYS.51	**ESCD_1.FA**	**EMIFSYS.50**	**ESCD_1.FB**	**EMIFSYS.49**	**ESCD_2.EF**	**EMIFSYS.48**
ESCD_1.FA	EMIFSYS.51	ESCD_1.FB	EMIFSYS.50	ESCD_2.EF	EMIFSYS.49	ESCD_2.F0
1995-07-17 MMF1013	1995-07-17 MMF1013	1995-07-17 MMF1012	1995-07-17 MMF1012	1995-07-17 MMF1011	1995-07-17 MMF1011	1995-07-17 MMF1010

ESCD_2.F0	**EMIFSYS.47**	**ESCD_2.F1**	**EMIFSYS.46**	**ESCD_2.F2**	**EMIFSYS.45**	**ESCD_2.F3**
EMIFSYS.48	ESCD_2.F1	EMIFSYS.47	ESCD_2.F2	EMIFSYS.46	ESCD_2.F3	EMIFSYS.45
1995-07-17 MMF1010	1995-07-17 MMF1009	1995-07-17 MMF1009	1995-07-17 MMF1008	1995-07-17 MMF1008	1995-07-17 MMF1007	1995-07-17 MMF1007

EMIFSYS.44	**ESCD_2.F4**	**EMIFSYS.43**	**ESCD_2.F5**	**EMIFSYS.42**	**ESCD_2.F6**	**EMIFSYS.41**
ESCD_2.F4	EMIFSYS.44	ESCD_2.F5	EMIFSYS.43	ESCD_2.F6	EMIFSYS.42	ESCD_2.F7
1995-07-17 MMF1006	1995-07-17 MMF1006	1995-07-17 MMF10066	1995-07-17 MMF10066	1995-07-17 MMF1005	1995-07-17 MMF1005	1995-07-17 MMF1004

ESCD_2.F7	**EMIFSYS.40**	**ESCD_2.F8**	**C101**	**ESCD_3.C4**
EMIFSYS.41	ESCD_2.F8	EMIFSYS.40	ESCD_3.C4	C101
1995-07-17 MMF1004	1995-07-17 MMF1003	1995-07-17 MMF1003	1995-07-17 MMF1234	1995-07-17 MMF1234

The following is a sample of cable labels as saved in a comma delimited ASCII file.

```
EMIFSYS.51,ESCD_1.FA,1995-07-17,MMF1013
ESCD_1.FA,EMIFSYS.51,1995-07-17,MMF1013
EMIFSYS.50,ESCD_1.FB,1995-07-17,MMF1012
ESCD_1.FB,EMIFSYS.50,1995-07-17,MMF1012
EMIFSYS.49,ESCD_2.EF,1995-07-17,MMF1011
ESCD_2.EF,EMIFSYS.49,1995-07-17,MMF1011
EMIFSYS.48,ESCD_2.F0,1995-07-17,MMF1010
ESCD_2.F0,EMIFSYS.48,1995-07-17,MMF1010
EMIFSYS.47,ESCD_2.F1,1995-07-17,MMF1009
ESCD_2.F1,EMIFSYS.47,1995-07-17,MMF1009
EMIFSYS.46,ESCD_2.F2,1995-07-17,MMF1008
ESCD_2.F2,EMIFSYS.46,1995-07-17,MMF1008
EMIFSYS.45,ESCD_2.F3,1995-07-17,MMF1007
ESCD_2.F3,EMIFSYS.45,1995-07-17,MMF1007
EMIFSYS.44,ESCD_2.F4,1995-07-17,MMF1006
ESCD_2.F4,EMIFSYS.44,1995-07-17,MMF1006
EMIFSYS.43,ESCD_2.F5,1995-07-17,MMF10066
ESCD_2.F5,EMIFSYS.43,1995-07-17,MMF10066
EMIFSYS.42,ESCD_2.F6,1995-07-17,MMF1005
ESCD_2.F6,EMIFSYS.42,1995-07-17,MMF1005
EMIFSYS.41,ESCD_2.F7,1995-07-17,MMF1004
ESCD_2.F7,EMIFSYS.41,1995-07-17,MMF1004
EMIFSYS.40,ESCD_2.F8,1995-07-17,MMF1003
ESCD_2.F8,EMIFSYS.40,1995-07-17,MMF1003
C101,ESCD_3.C4,1995-07-17,MMF1234
ESCD_3.C4,C101,1995-07-17,MMF1234
                                    Cable ID
                                    Date when assigned
                                    Target (end point B)
                                    Source (end point A)
```
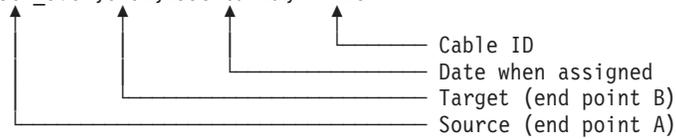
Chapter 13. Scenarios

The scenarios in this chapter have been designed to represent typical procedures that you would go through in the course of maintaining your configuration. They cover the following tasks:

- "Scenario I – Adding and connecting new equipment"
- "Scenario II – Defining a CTC connection" on page 284
- "Scenario III – Defining a fiber cabling infrastructure" on page 295
- "Scenario IV – Configuration changes for installing a FICON bridge card in a 9032–5 ESCON director" on page 303
- "Scenario V – Establish a PPRC connection consisting of several PPRC links" on page 307

Scenario I – Adding and connecting new equipment

In this scenario we will create new CHPIDs and strings for a processor with two partitions and connect them to each other through a new controller, a new ESCON switch, and a patchport cabinet. An operation system configuration already exists.

Add four new channels to an existing LPAR processor

1. Edit the processor by selecting **Edit:Processor**. This opens the **Edit a Processor** dialog.
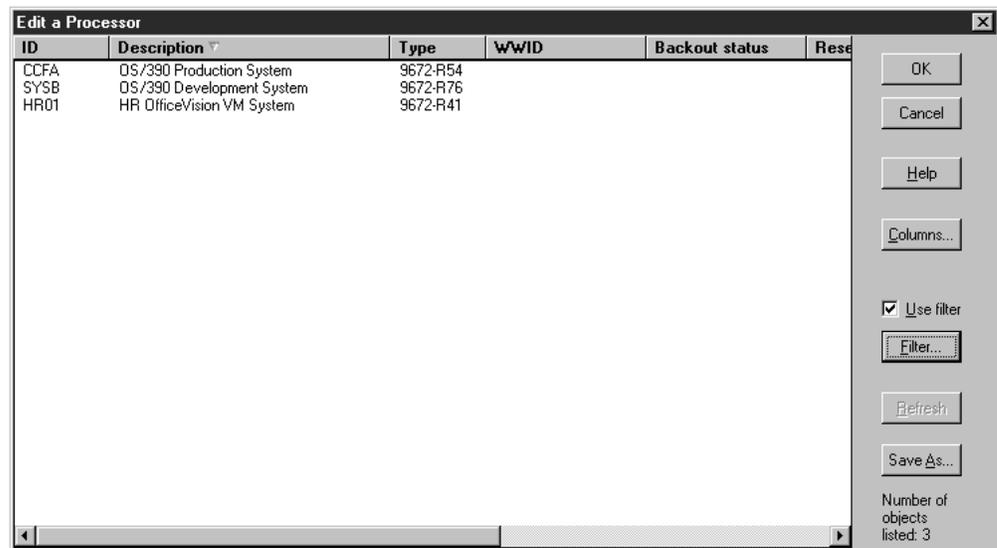
2. Choose the processor from the list (in our example, *SYSB*) and press **OK** or double-click on the processor in the list. This opens the **Processor** dialog.

Processor

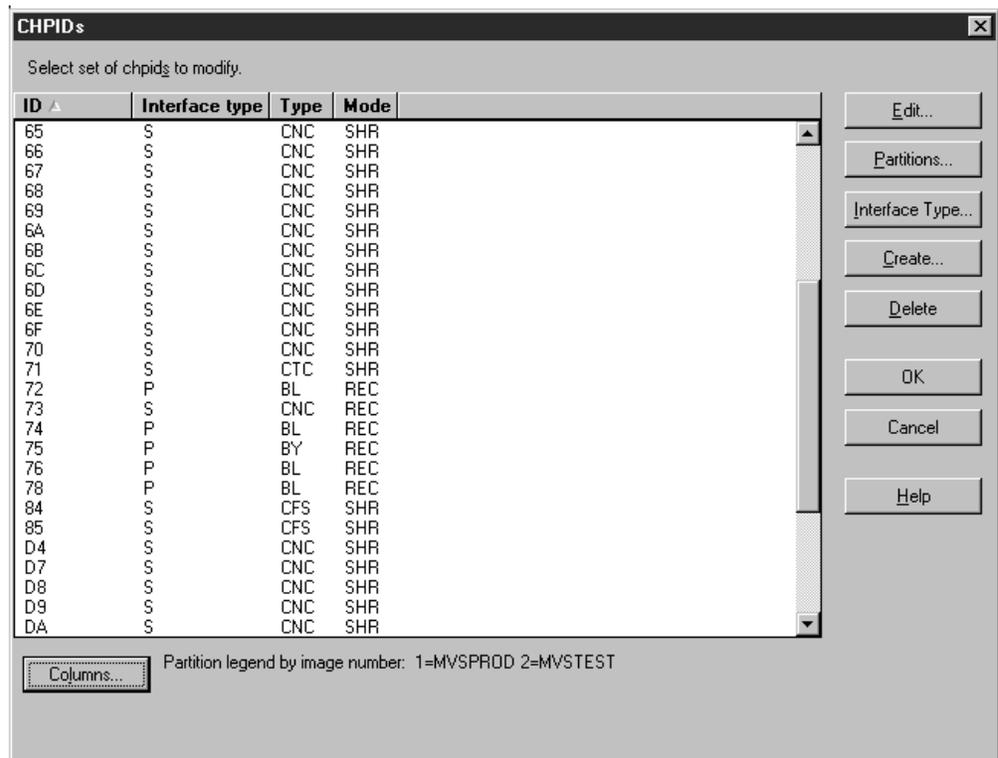ID: SYSB   Short name: SY

System Name:

Description: OS/390 Development System

Serial No.: 0000019672

Type: 9672

Model: R76

Mode: LPAR

Support Level: FCV, OSD, OSE, IC, ICB, CF, OSA, ESCON, Parallel

Partitions...
CHPIDs...
Edit...
Status...
OK
Cancel
Help

SNA address (S/390 microprocessor cluster)

Network name:

CPC name:

---
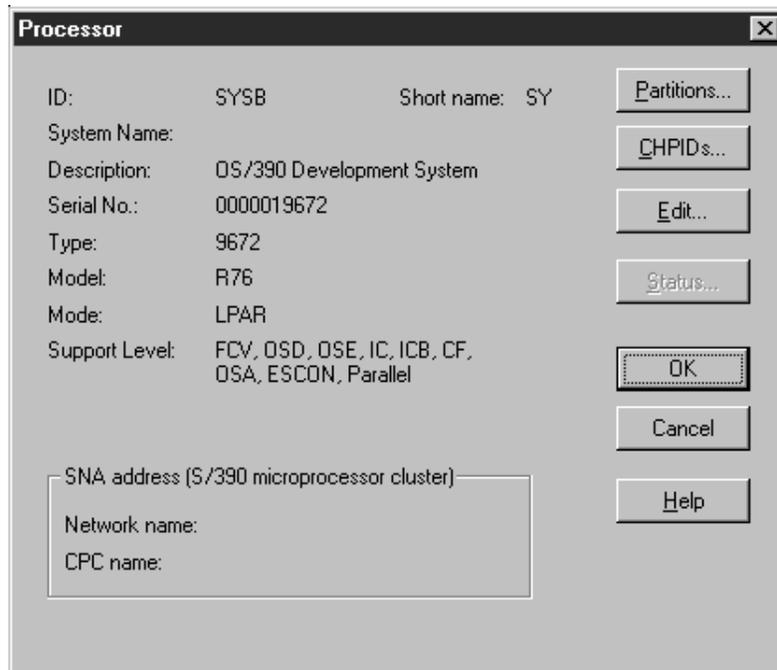
3. Click **CHPIDs...**. This opens the **CHPIDs** dialog.

CHPIDs

Select set of chpids to modify.

ID △	Interface type	Type	Mode
65	S	CNC	SHR
66	S	CNC	SHR
67	S	CNC	SHR
68	S	CNC	SHR
69	S	CNC	SHR
6A	S	CNC	SHR
6B	S	CNC	SHR
6C	S	CNC	SHR
6D	S	CNC	SHR
6E	S	CNC	SHR
6F	S	CNC	SHR
70	S	CNC	SHR
71	S	CTC	SHR
72	P	BL	REC
73	S	CNC	REC
74	P	BL	REC
75	P	BY	REC
76	P	BL	REC
78	P	BL	REC
84	S	CFS	SHR
85	S	CFS	SHR
D4	S	CNC	SHR
D7	S	CNC	SHR
D8	S	CNC	SHR
D9	S	CNC	SHR
DA	S	CNC	SHR

Edit...
Partitions...
Interface Type...
Create...
Delete
OK
Cancel
Help

Columns... Partition legend by image number: 1=MVSPROD 2=MVSTEST

---

4. Click **Create...**. This opens the **Create CHPIDs** dialog.
5. Fill in the fields as shown, specifying appropriate CHPID numbers, to create four new shared (SHR) CNC CHPIDs:
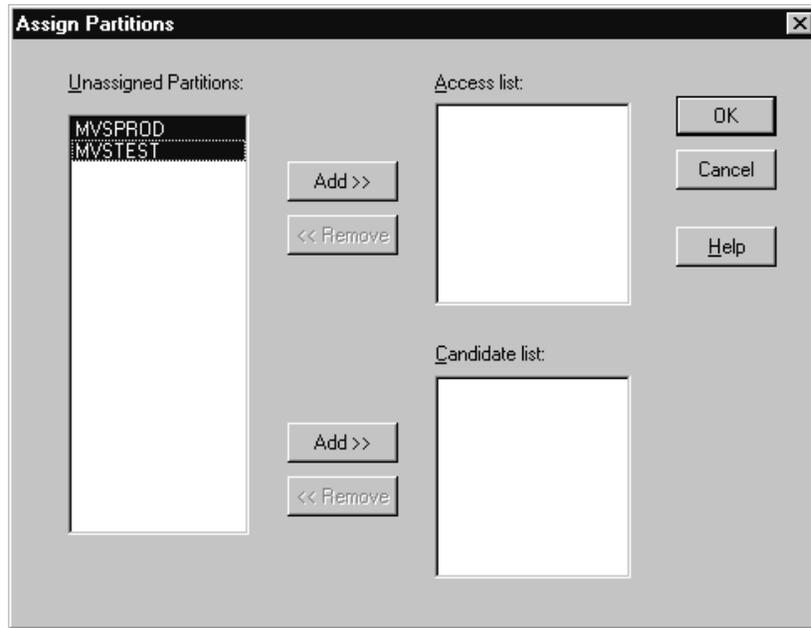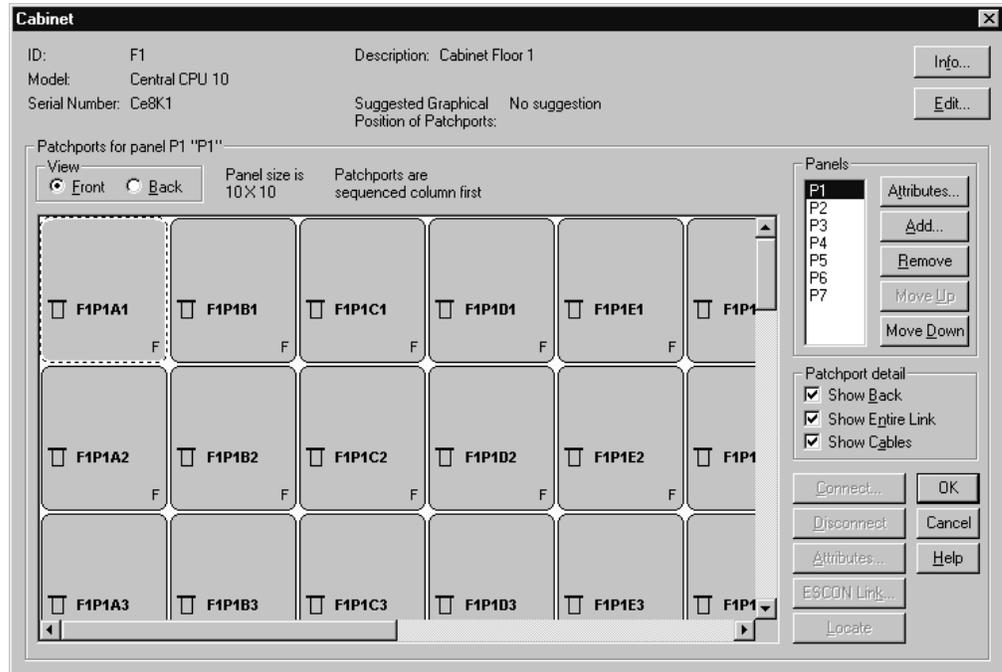
6. Click **OK**. Since the processor is in LPAR mode, this opens the **Assign Partitions** dialog to choose which partitions to assign to the access and candidate lists of the new CHPIDs.
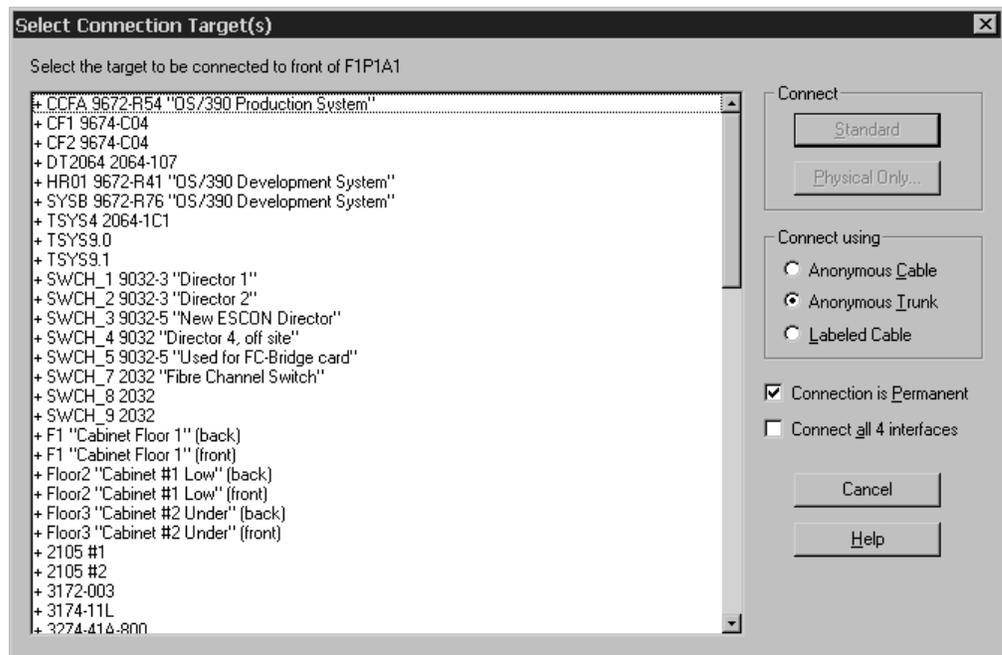
7. Select all of the partitions and click **Add>>** beside the access list. This adds the selected partitions to the access list of the four CHPIDs.

8. Click **OK** on all open dialogs (besides the **Edit a Processor** dialog) to close them. On the **Edit a Processor** dialog click **Close**. The new CHPIDs have now been added.

9. Select **View:Filter diagram** and select the **Show All CHPIDs** checkbox to force the new CHPIDs to be included in the diagram.

Connect the new CHPIDs to patchports

1. Edit an existing patchport cabinet by either selecting **Edit:Cabinet** and choosing the cabinet from the selection list or by double-clicking on the cabinet in the diagram. This opens the **Cabinet** dialog.

2. Select a panel from the **Panels** listbox (in our example, *P1*).

Cabinet

ID: F1 Description: Cabinet Floor 1 Info...
Model: Central CPU 10 Edit...
Serial Number: Ce8K1 Suggested Graphical No suggestion
Position of Patchports:

Patchports for panel P1 "P1"

View ● Front ○ Back Panel size is 10×10 Patchports are sequenced column first

Panels
P1 Attributes...
P2
P3 Add...
P4
P5 Remove
P6
P7 Move Up
Move Down

F1P1A1 F1P1B1 F1P1C1 F1P1D1 F1P1E1 F1P1
F F F F F

F1P1A2 F1P1B2 F1P1C2 F1P1D2 F1P1E2 F1P1
F F F F F

F1P1A3 F1P1B3 F1P1C3 F1P1D3 F1P1E3 F1P1

Patchport detail
☑ Show Back
☑ Show Entire Link
☑ Show Cables

Connect... OK
Disconnect Cancel
Attributes... Help
ESCON Link...
Locate

3. Find four consecutive free patchports on the diagram of the patchport panel and select them. You may have to scroll the panel diagram.

4. Click **Connect**. This opens the **Select Connection Target(s)** dialog. In the listbox, double-click on the processor with the new CHPIDs (*SYSB 9672-R76*). This expands the list, showing all CHPIDs in the processor.

Select Connection Target(s)

Select the target to be connected to front of F1P1A1

+ CCFA 9672-R54 "OS/390 Production System"
+ CF1 9674-C04
+ CF2 9674-C04
+ DT2064 2064-107
+ HR01 9672-R41 "OS/390 Development System"
+ SYSB 9672-R76 "OS/390 Development System"
+ TSYS4 2064-1C1
+ TSYS9.0
+ TSYS9.1
+ SWCH_1 9032-3 "Director 1"
+ SWCH_2 9032-3 "Director 2"
+ SWCH_3 9032-5 "New ESCON Director"
+ SWCH_4 9032 "Director 4, off site"
+ SWCH_5 9032-5 "Used for FC-Bridge card"
+ SWCH_7 2032 "Fibre Channel Switch"
+ SWCH_8 2032
+ SWCH_9 2032
+ F1 "Cabinet Floor 1" (back)
+ F1 "Cabinet Floor 1" (front)
+ Floor2 "Cabinet #1 Low" (back)
+ Floor2 "Cabinet #1 Low" (front)
+ Floor3 "Cabinet #2 Under" (back)
+ Floor3 "Cabinet #2 Under" (front)
+ 2105 #1
+ 2105 #2
+ 3172-003
+ 3174-11L
+ 3274-41A-800

Connect
Standard
Physical Only...

Connect using
○ Anonymous Cable
● Anonymous Trunk
○ Labeled Cable

☑ Connection is Permanent
☐ Connect all 4 interfaces

Cancel
Help

5. Select a CHPID and the "Connect all 4 interfaces" checkbox. Choose the type of connection you want to use and whether the connection should be permanent.

6. The patchports will be connected to four consecutive CHPIDs, starting with the one you have just selected.

7. Close all open dialogs.
8. Select the **View:Filter Diagram** menu item and select the **Objects Connected to Processors but not to Controllers** and the **Show Patchports** checkbox. The patchports connected to the new CHPIDs will appear in the diagram.

Add a new ESCON switch

1. Select **Create:Switch**. This opens a **Create Switch** dialog.
2. Fill in the dialog, specifying appropriate control unit and device numbers, and click **OK**. The new switch will appear selected in the diagram.
3. Edit the new switch by either selecting **Edit:Switch** and choosing the switch from the list or by double-clicking on the switch in the diagram. This opens the **Switch** dialog.
4. Before connecting switch ports to the CHPIDs, we want to mark all ports as installed. Click **Edit**. This opens the **Edit Switch** dialog.

5. Click **Edit** in the **Ports** box. This opens the **Switch Ports** dialog.
6. Select the ports you want to install and click **Install**. The selected ports are now installed.
7. Click on the **Close** button, then **OK** on the **Edit Switch** dialog to return to the **Switch** dialog.
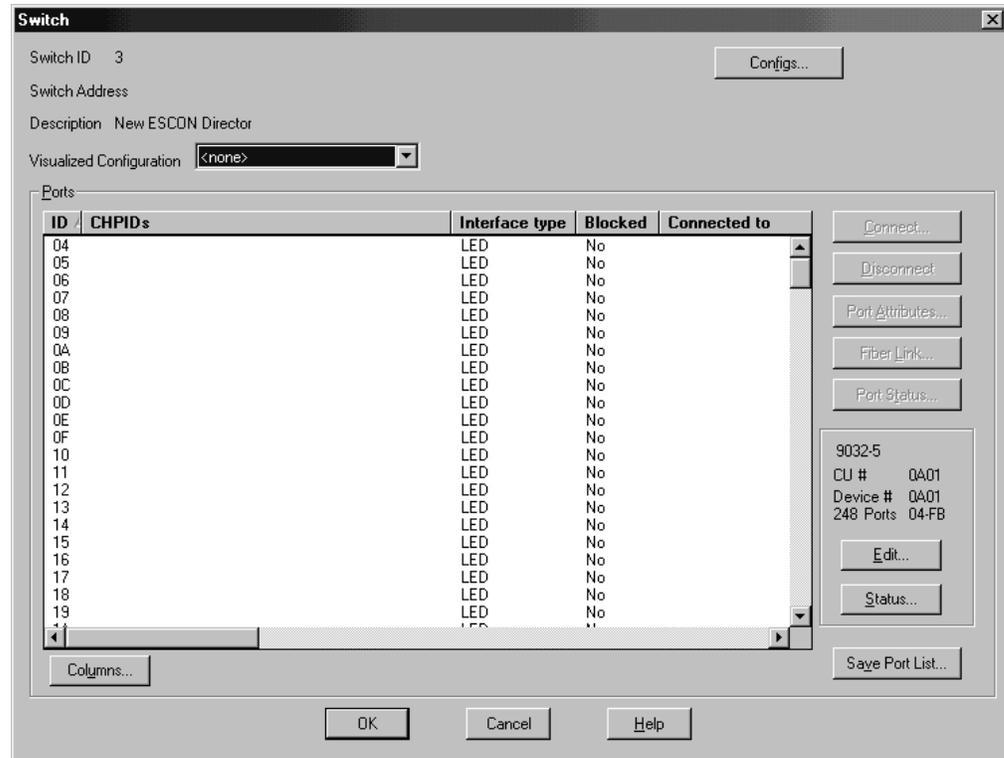
Connect the new switch to one of the new CHPIDs

Although the new CHPIDs are already connected to patchports in the cabinet, we specify the CHPID as the target for the connection. The patchport will automatically be connected between the switch port and the CHPID.

1. With the **Switch** dialog still open, select one of the switch ports and click **Connect**. This opens the **Connect Switch Port** dialog.
2. In the list box, double-click on the processor (in our example SYSB) with the four new CHPIDs. This expands the list, showing all CHPIDs in the processor.
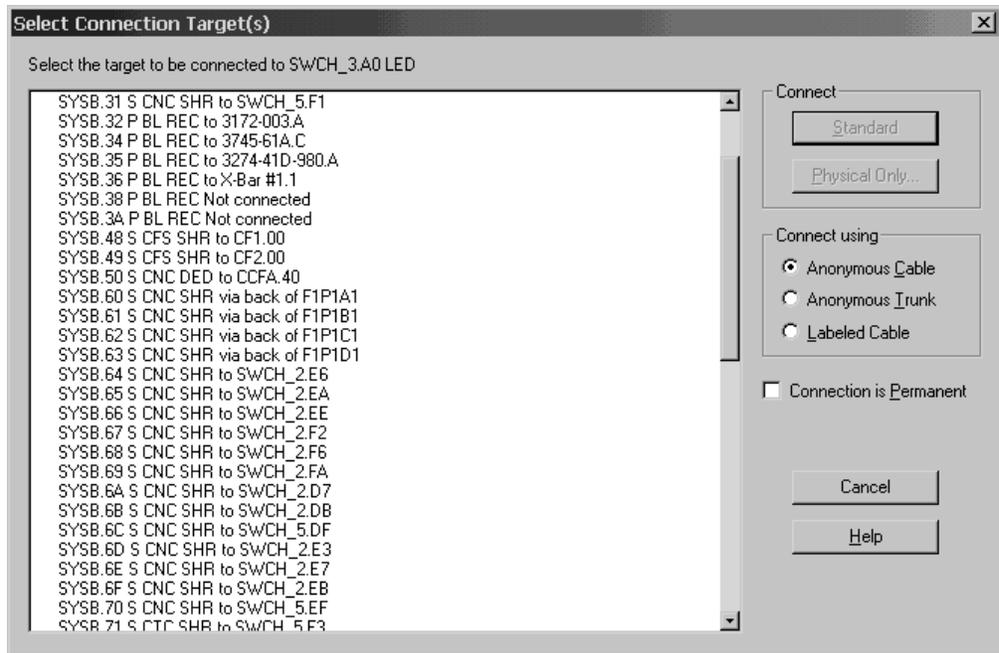
3. Double-click on the first of the four new CHPIDs. The connection is made between the switch port and the CHPID.

4. Click **OK** on all open dialogs to close them. The new connection between the switch and the new CHPID can be seen on the diagram.

Connect another switch to the second new CHPID

Follow the procedure above to connect another switch to the second new CHPID. You may have to create the second switch following the procedure above. If there are no free installed switch ports you may have to mark some ports installed as described above.

Add a new controller

We will be adding a 3990 DASD subsystem.

1. Select **Create:Controller**. This opens the **Create Controller** dialog. Fill in the values as shown, specifying an unused control unit number, and click **OK**.
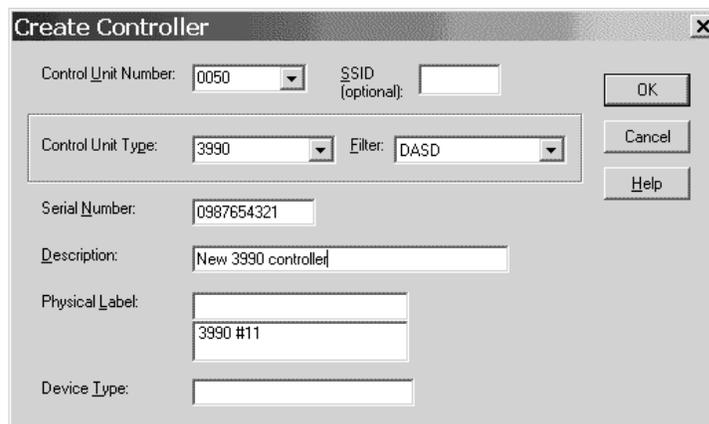
2. The **Choose Physical Description** dialog opens. The dialog is displayed when there is more than one physical description possible for the control unit type selected. Select the description as shown and click **OK**.
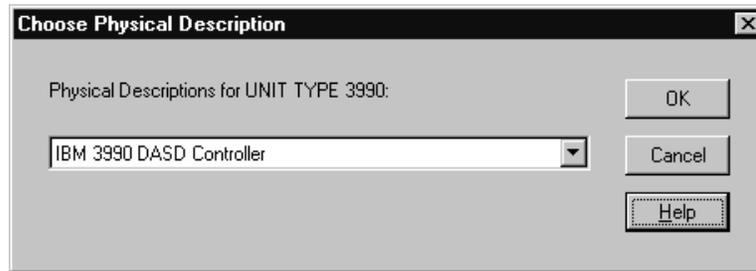
3. Since a 3990 has two storage clusters (generically called "segments" in HCM), the **Add Control Unit** dialog is opened to create a second control unit for the second segment.
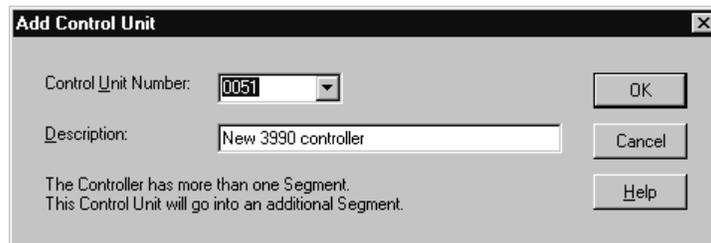
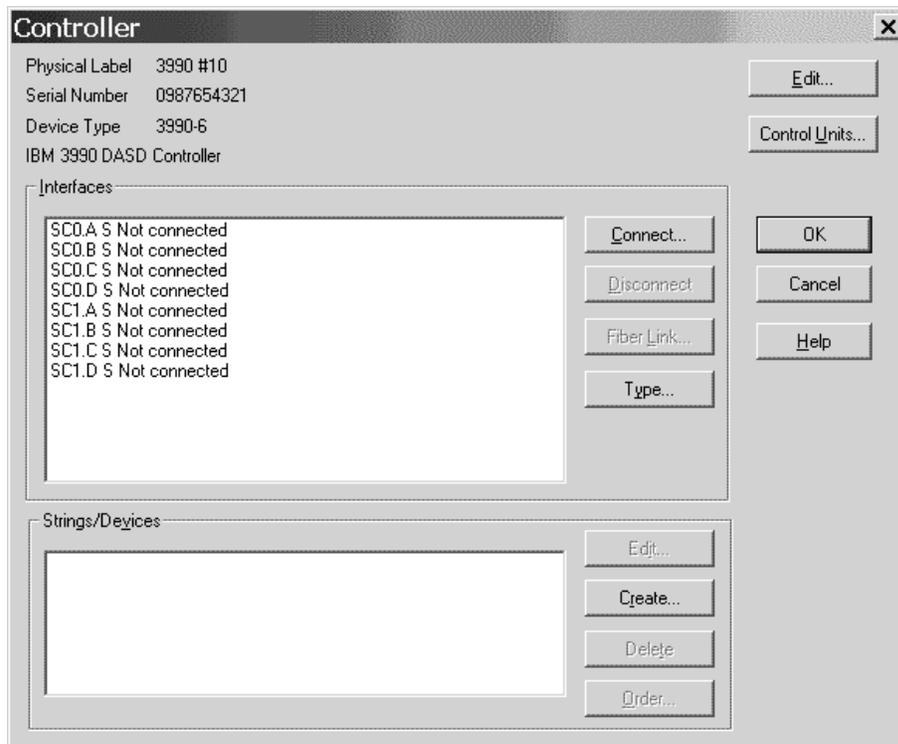4. Enter an unused control unit number and click **OK**.
5. The new controller appears selected on the diagram.

Connect the controller to CHPIDs

There are two ways to connect a string of devices to a processor with HCM. One way is to connect the controller(s) to the CHPID(s) first (possibly through switches and patchport cabinets), then connect the strings to the controllers. We call this "connecting top-down". Alternately, you can connect the strings to the controllers first, then connect the controllers to the CHPID(s). We call this "connecting bottom-up".
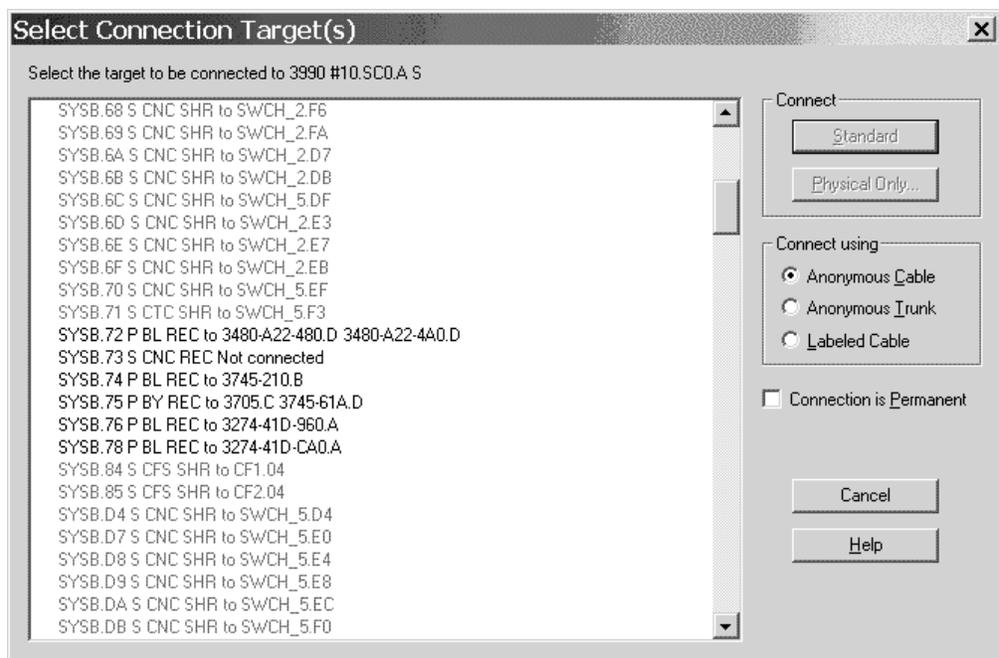
The advantage of connecting top-down is that you are given immediate feedback about available unit addresses when you connect strings to controllers. The advantage of connecting bottom-up is that when you connect a controller with strings to a CHPID, only the unit address ranges that are used by the connected devices are defined for the controller, instead of the default unit address range of 00 through 64.

We will be connecting top-down, starting with connecting one channel interface on each segment of the new controller to a CHPID.

1. Edit the new controller by either selecting **Edit:Controller** and choosing the controller from the list or by double-clicking on the controller in the diagram. This opens the **Controller** dialog.

Controller

Physical Label 3990 #10
Serial Number 0987654321
Device Type 3990-6
IBM 3990 DASD Controller

Interfaces

```
SC0.A S Not connected
SC0.B S Not connected
SC0.C S Not connected
SC0.D S Not connected
SC1.A S Not connected
SC1.B S Not connected
SC1.C S Not connected
SC1.D S Not connected
```

Connect...
Disconnect
Fiber Link...
Type...

Edit...
Control Units...

OK
Cancel
Help

Strings/Devices

Edit...
Create...
Delete
Order...

2. Select a free channel interface on the first segment (for example, SC0.A) and click **Connect**. This opens the **Select Connection Target(s)** dialog.

Select Connection Target(s)

Select the target to be connected to 3990 #10.SC0.A S

```
SYSB.68 S CNC SHR to SWCH_2.F6
SYSB.69 S CNC SHR to SWCH_2.FA
SYSB.6A S CNC SHR to SWCH_2.D7
SYSB.6B S CNC SHR to SWCH_2.DB
SYSB.6C S CNC SHR to SWCH_5.DF
SYSB.6D S CNC SHR to SWCH_2.E3
SYSB.6E S CNC SHR to SWCH_2.E7
SYSB.6F S CNC SHR to SWCH_2.EB
SYSB.70 S CNC SHR to SWCH_5.EF
SYSB.71 S CTC SHR to SWCH_5.F3
SYSB.72 P BL REC to 3480-A22-480.D 3480-A22-4A0.D
SYSB.73 S CNC REC Not connected
SYSB.74 P BL REC to 3745-210.B
SYSB.75 P BY REC to 3705.C 3745-61A.D
SYSB.76 P BL REC to 3274-41D-960.A
SYSB.78 P BL REC to 3274-41D-CA0.A
SYSB.84 S CFS SHR to CF1.04
SYSB.85 S CFS SHR to CF2.04
SYSB.D4 S CNC SHR to SWCH_5.D4
SYSB.D7 S CNC SHR to SWCH_5.E0
SYSB.D8 S CNC SHR to SWCH_5.E4
SYSB.D9 S CNC SHR to SWCH_5.E8
SYSB.DA S CNC SHR to SWCH_5.EC
SYSB.DB S CNC SHR to SWCH_5.F0
```

Connect

Standard
Physical Only...

Connect using

- Anonymous Cable
- Anonymous Trunk
- Labeled Cable

☐ Connection is Permanent

Cancel
Help

3. Choose the desired processor (in our example SYSB) from the list and double-click it to expand it. This lists all of the CHPIDs in the processor.

4. Select a SHR CNC CHPID with no connection and click **Standard** in the **Connect** box. A *standard connection* makes the physical connection and all appropriate logical definitions, using standard values.

The connection is now made including all definitions between the control unit and the CHPID and its processor. Had there been devices connected to the control unit, definitions would also have been made between the devices and the partitions and the processor.

5. You are now back in the **Controller** dialog. Select a free channel interface on the second segment (e.g. SC1.A) and connect it to another SHR CNC CHPID following the procedure above.

6. Click **OK** on all open dialogs to close them. The connections between the controller and the CHPIDs appear on the diagram. When the controller is selected, the full connection is highlighted from the controller to the CHPIDs, and to any partitions that can access the CHPIDs.

Connect the controller to the ESCON switches

We will connect a second channel interface on each segment of the new controller through a switch and on to the new CHPIDs.

1. Open the **Controller** dialog for the new controller as above.

2. Select a free channel interface on the first segment (e.g. SC0.B) and click **Connect**. This opens the **Select Connection Target(s)** dialog.

3. Choose the new switch from the list and double-click it to expand it. This lists all of the installed ports in the new switch.
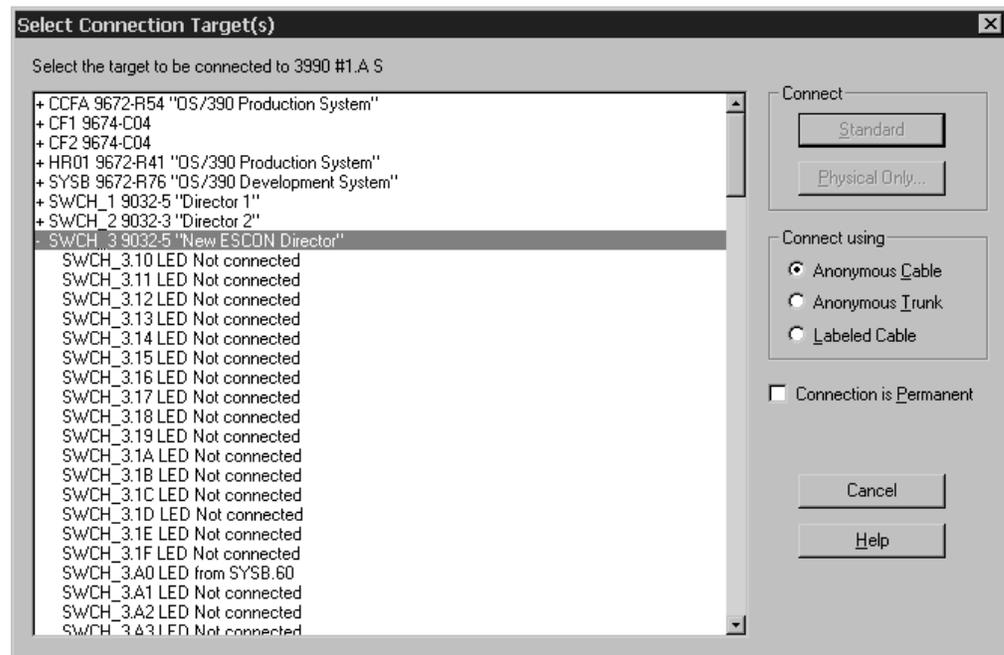
4. Select a port with no connection and click **Standard** in the **Connect** box. A *standard connection* makes the physical connection and all appropriate logical definitions, using standard values.

5. The **Processor and CHPID Connections** dialog is opened, where you can specify the CHPIDs to which the control unit will be defined.

Processor and CHPID Connections   [?][X]

SYSB | Summary

SYSB CHPIDs connected to control unit 0050 via 3990 #10.SC0.A

Reachable CHPIDs

CHPID	Type	Mode	Port	Dynamic director	Device count
60	CNC	SHR	03.A0	03	0

Add >

< Remove

Show
☐ Partitions
☐ Control units

Sort
⦿ By CHPID   ○ By Device count

Connected CHPIDs

CHPID	Type	Mode	Port	Dynamic director	DLA	Device count

No. of Managed CHPIDs [0 ▼]   Attributes...

OK   Cancel   Help
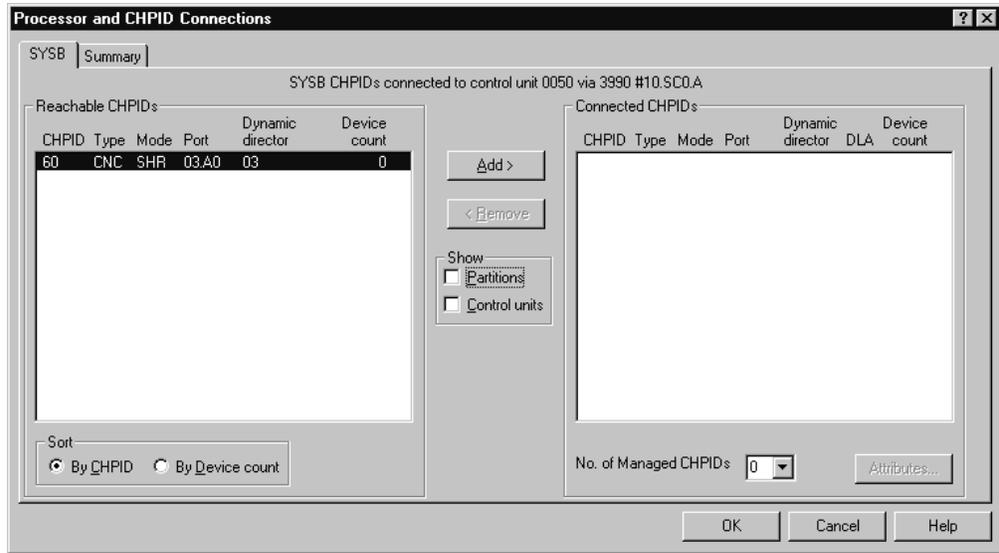
6. In the **Processor and CHPID Connections** dialog, from the tabs first select the processor you want the control unit to connect with. Then select the desired CHPID(s), click **Add** to move them to the **Connected CHPIDs** list, and click **OK**.

   The connection is now made including all definitions among the control unit, the switch ports, the CHPID and the processor. Had there been devices connected to the control unit, the definitions would also have been made between the devices and the partitions and the processor.

7. You are now in the **Controller** dialog. Select a free channel interface on the other segment (e.g. SC1.B) and connect it through the second switch to the second new CHPID following the procedure above.

8. Click **OK** on all open dialogs to close them. The connections between the controller and the switches appear on the diagram. When the controller is selected, the full connection is highlighted from the controller through the switches and the patchports, to the CHPIDs, and finally to the partitions.

Define the fiber links for the connections

When a connection is made between two pieces of equipment no cable is initially assigned to the connection. You can assign cables, complete with labels, types, and lengths, to the connections in your configuration. This information can then appear on the diagram and can be printed in reports. If the connection is between two fiber transceivers, you can add patchports to the fiber link.

We will route one of the connections we just made between the new controller and a switch, passing through a patchport cabinet, and label the cables involved.

1. Edit the new controller.

2. In the **Controller** dialog, select (do not double-click) one of the two channel interfaces that are connected to switches. Because this is a fiber interface, the **Fiber Link** button is enabled.

3. Click **Fiber Link**. The **Edit Fiber Link** dialog is opened. It shows the composition of the fiber link in the box to the right. Right now the only thing between the channel interface and the switch port is a connection with no cable, represented by “- - -”.

4. Click (do not double-click) on the connection (the "- - -") to indicate where we want to insert a patchport. This causes the list box on the left side to list all patchport cabinets in the configuration.

5. Choose the desired cabinet and double-click on it. This expands the list to show all patchports in the cabinet.
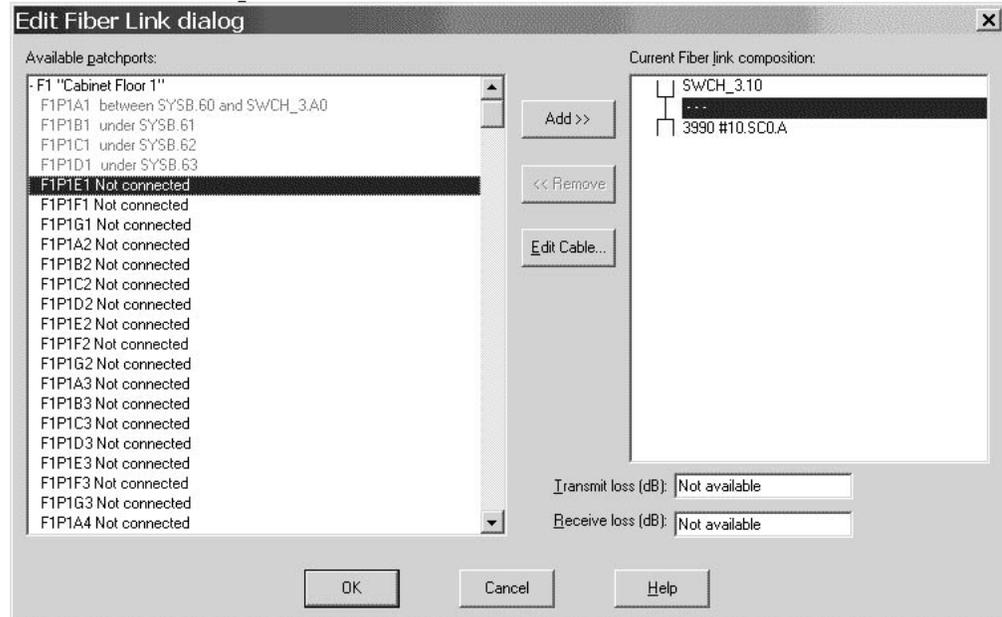
6. Select a free patchport and click **Add>>**.

7. The **Add Patchport to Fiber Link** dialog appears. You can decide whether you want to connect the cables to the front or to the back side of the patchport. This inserts the patchport into the fiber link such that there are now two unlabeled cables on either side of it. Press **OK**.

8. Now we will label one of the cables. Select the "- - -" between the controller and the patchport and click **Edit Cable**. This opens the **Assign Cable** dialog.

9. Fill in the cable's ID, type, length, and optionally the Comment field, or select an existing cable from the unused cables list. Then, click **OK**. The cable's new label now appears in the **Fiber Link** dialog.

10. Label the other cable in the same way.

11. Click **OK** to return to the **Controller** dialog. You can edit the fiber links of the other connected channel interfaces by following the same procedure.

12. Click **OK** on all open dialogs to close them.

13. Use the **Locate:Controller** menu item to zoom in on the new controller. Select **View:Text Detail** and check the **Cable Description** checkbox. The cable labels will now appear on the diagram.

Add some strings to the new controller

1. Select **Create:String**. This opens the **Create String** dialog.

2. Fill in the **Create String** dialog as shown, specifying an appropriate device number.
3. Since there is more than one physical description for this unit type, the **Choose Physical Description** dialog is opened. Choose the appropriate physical description and click **OK**.

```
Choose Physical Description                              [x]

Physical Descriptions for UNIT TYPE 3390:          [  OK  ]

[IBM 3390 all models                          |▼]  [ Cancel ]

                                                   [  Help  ]
```
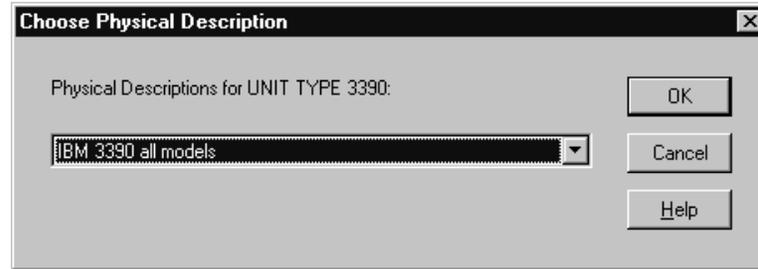
4. This creates a string of 32 devices. Double-click on the new string on the diagram to open the **String** dialog. (Tip: position the mouse cursor in the top rectangle of the string.)

```
String                                                  [x]

 EMS database                              [   Edit...    ]

    IBM 3390 all models
                                           [  I/O Devices... ]
 Number of devices:   32

 ┌Interfaces──────────────────────────────┐
 │A Not connected                        │ [  Connect...  ]
 │B Not connected                        │
 │C Not connected                        │ [  Disconnect  ]
 │D Not connected                        │
 └────────────────────────────────────────┘

       [   OK   ]     [  Cancel  ]     [   Help   ]
```
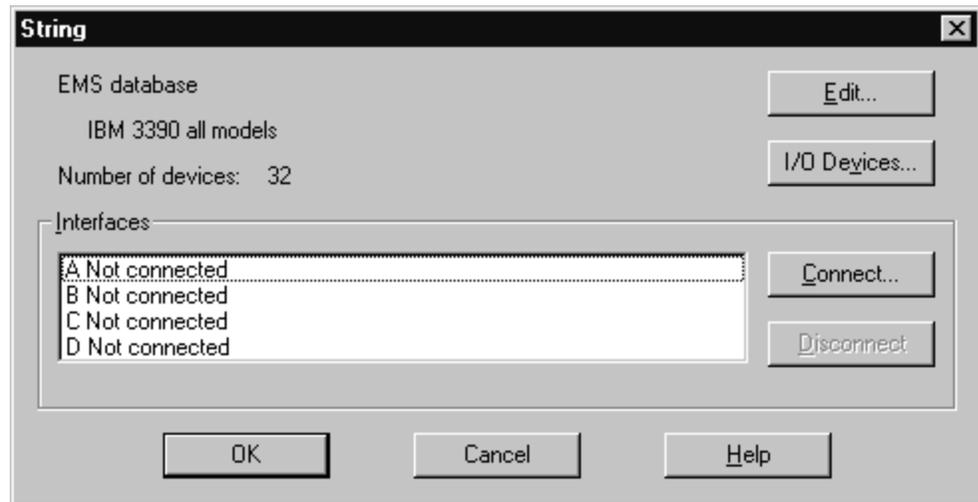
5. Select the first string interface (A) and click **Connect**. This opens the **Select Controller Device Interface** dialog.

Select Controller Device Interface for EMS database

Choose a connection target for EMS database.A

```
- 3990 #1 Not connected
    3990 #1.SC0.SP0 Not connected
    3990 #1.SC0.SP1 Not connected
    3990 #1.SC1.SP2 Not connected
    3990 #1.SC1.SP3 Not connected
+ 3990-003-200 reaching String 0240-025F String 0260-027F ...
+ 3990-003-240 reaching String 0200-021F String 0220-023F ...
+ 3990-003-360 reaching String 0360-037F disk340
+ 3990-003-500 reaching String 500
+ 3990-003-660 reaching String 660
+ 3990-006-280 reaching RAMAC280
+ 3990-006-300 reaching RAMAC300
+ 3990-006-380 reaching String 0380-039F String-3A0
+ 3990-006-600 reaching RAMAC600
+ NSCLCA1 #1 reaching String 0B00-0B1F String 0B20-0B3F
```

Connect

| Standard | Custom... | Physical Only | | Cancel | Help |

6. Double-click on the new controller and choose the storage processor (generically called a "device interface" in HCM) to connect to and double-click on it.

   The connection is now made including all definitions between the devices and the control unit, CHPIDs, partitions, and processor.

7. In the **String** dialog, select the remaining string interfaces (B, C, and D) and click **connect**. This will open the **Select Controller Device Interface** dialog three times in sequence to connect the string interfaces to the remaining device interfaces of the new controller.

8. Create a second string of 3390 devices and also connect it to the new controller using the same procedure. The second string will be connected after the first one on the daisy-chain.

9. Click **OK** on all open dialogs to close them. The connections between the string and the controller appear on the diagram. When the string is selected, the full connection is highlighted from the string all the way to the partitions.

Define the devices to the operating system

The newly created devices still need to be defined to the appropriate operating system.

1. Edit one of the new strings by either selecting **Edit:String** and choosing the string from the list or by double-clicking on the string in the diagram. This opens the **String** dialog.

2. Click **I/O Devices**. This opens the **I/O Device List** dialog. By default all devices in the string are selected so that we can define them all to the operating system at once.
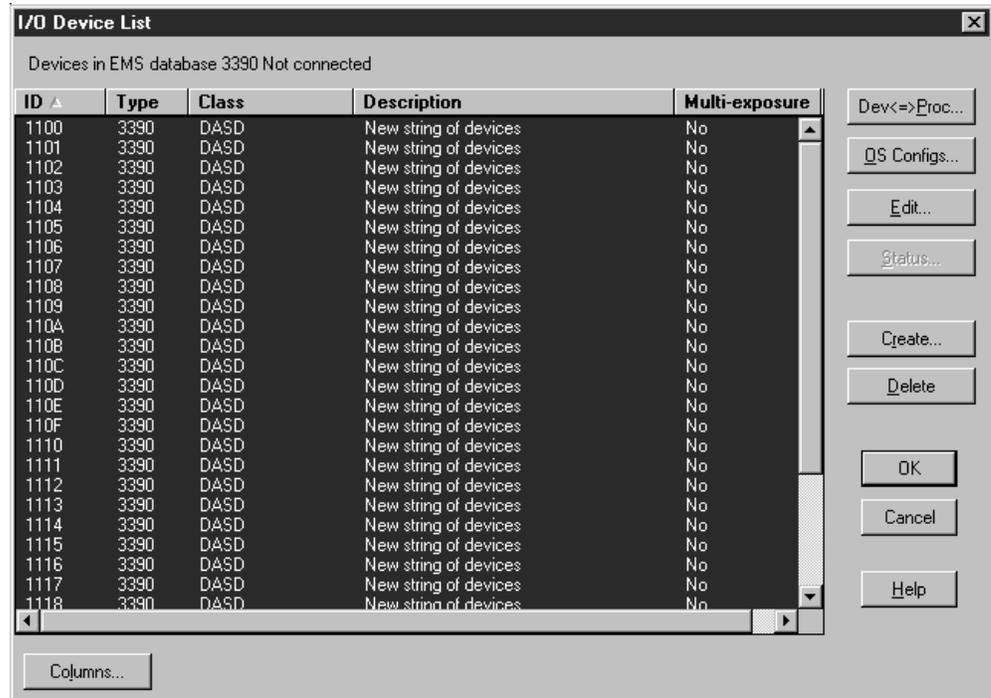
3. Click **OS Configs**. This opens a dialog that lists all operating system configurations in the IODF.
4. Choose the operating system configuration to which you want to define these devices and click **Define**. This opens the **Define Device <==> OS Configuration Parameters** dialog that lists all of the parameters that determine the device <==> OS definitions.
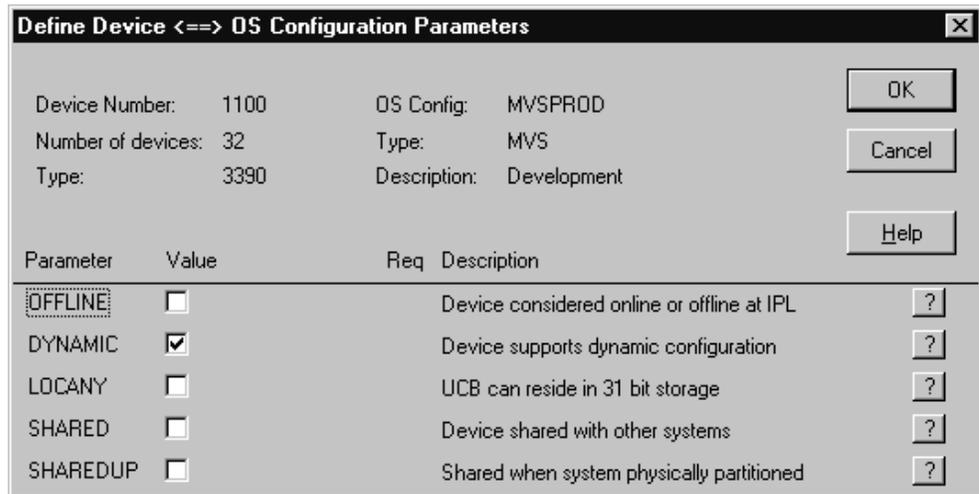
5. Set the parameters appropriately and click **OK**. This defines the devices to the selected OS configuration.
6. Click **OK** or **Close** on all open dialogs to close them.
7. Save your configuration file.

You have now completely installed and connected the new strings and the controller through the new switch.

Scenario II – Defining a CTC connection

In this scenario we create CTC connections involving three processors routed through a switch. We assume that the following processors and unconnected CHPIDs exist:

Processor PROC1 (LPAR mode with partitions LPAR11 and LPAR12)
CHPIDS:
>   20 CTC SHR (both partitions in the access list)
>
>   21 CNC SHR (both partitions in the access list)

Processor PROC2 (LPAR mode with partitions LPAR21 and LPAR22)

CHPIDS:
>   20 CTC DED (LPAR21 in the access list)

Processor PROC3 (Basic mode)

CHPIDS
>   21 CNC DED

We also assume that an operating system exists and that an ESCON Director 01 (9032) exists with ports E0-E3 free.

We will create four CTC connections:

```
PROC1.20  CTC  to  PROC1.21  CNC  (wraparound)
PROC2.20  CTC  to  PROC1.21  CNC
PROC1.20  CTC  to  PROC3.21  CNC
PROC2.20  CTC  to  PROC3.21  CNC
```

Connect a switch to the CHPIDs

1. Edit the switch by either selecting **Switch** from the **Edit** menu and choosing the switch from the list or by double-clicking on the switch in the diagram. This opens the **Switch** dialog.
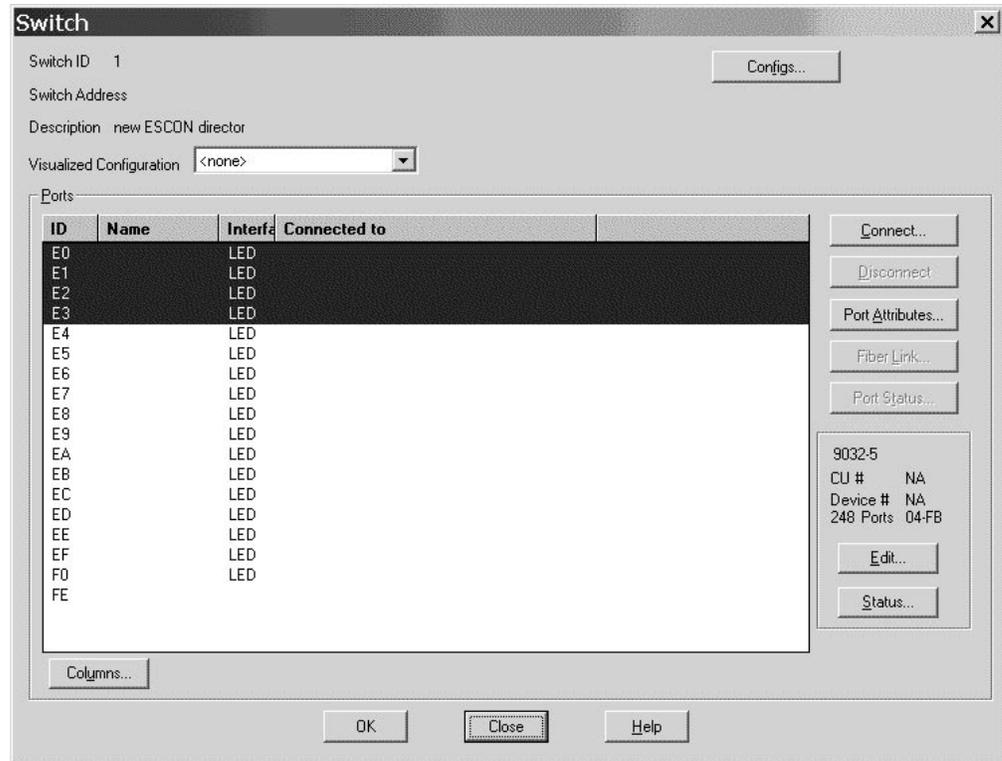
2. Select four free ports and click **Connect**. This will open the **Select Connection Target(s)** dialog four times in sequence, once for each port.
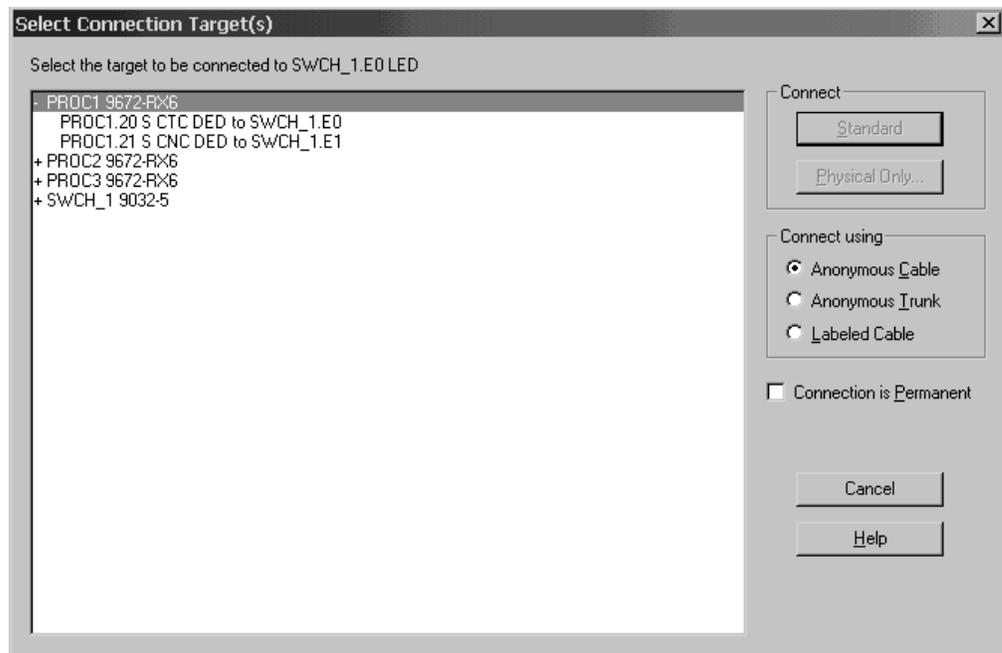
3. In the **Select Connection Target(s)** list, double-click on processor PROC1 to expand the list, showing all CHPIDs in the processor.
4. Double-click on CHPID PROC1.20 CTC SHR. The connection is made between the switch port and the CHPID.

5. The **Select Connection Target(s)** list will reappear three more times, allowing you to connect CHPIDs PROC1.21, PROC2.20 and PROC3.21 to the switch (ports E1 – E3).
6. Close the **Switch** dialog and select **Filter diagram** from the **View** menu. Enable the **Show All CHPIDs** item. The new connections will appear on the diagram.

Create the first CTC connection

HCM considers a CTC connection as a single entity comprising selected control units and I/O devices connected to the CHPIDs.

1. Select **CTC connection...** from the **Create** menu. This opens the **Create CTC Connection** dialog.
2. Click the **CTC or FC CHPID** button and select PROC1.20 CTC SHR. Even though the CHPIDs are connected to switch port, we select the CHPIDs that are to be connected. Click the second **CNC/FCV or FC CHPID** button and select PROC1.21 CNC SHR.
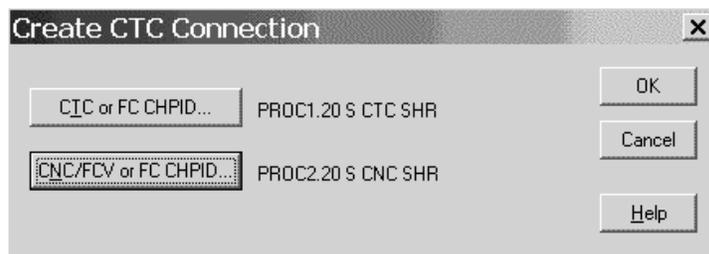
3. Click **OK**. The **Edit CTC Connection** dialog will automatically appear to allow you to define the control units that are part of the connection.
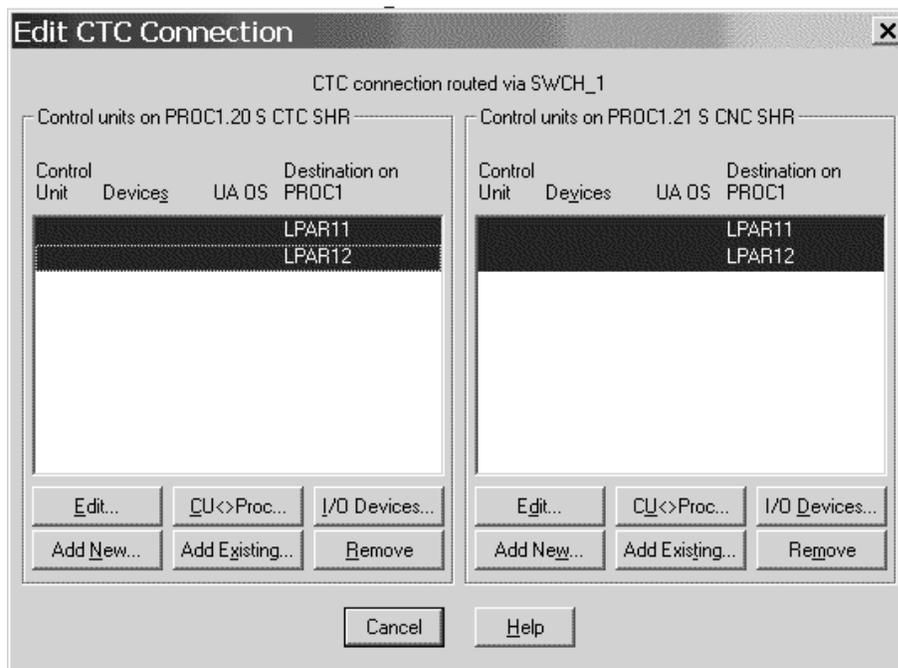
4. Click the **Add New...** button on the left-hand side to create the control units and devices that will be connected to PROC1.20 CTC to represent the images available via PROC1.21 CNC.

5. Enter control unit number 4110 to represent LPAR11 and control unit number 4120 to represent LPAR12. Leave the other controls as they appear and click **OK**. This will create control unit 4110 CTC with devices 4110,4 CTC and control unit 4120 CTC and devices 4120,4 CTC and connect them to CHPID PROC1.20 CTC so as to represent the two partitions.

```
Create CTC Control Units/Devices                                    [X]

    Create CTC Control Unit(s) to be connected to PROC1.20 S CTC SHR

            Number    Starting
            of        Device        Destination          ┌──────────┐
  Control   Devices   Address*      on PROC1             │    OK    │
  Unit                                                    └──────────┘
  [4110]    [4]       [    ]         LPAR11               ┌──────────┐
                                                          │  Cancel  │
  [4120]    [4]       [    ]         LPAR12               └──────────┘
                                                          ┌──────────┐
  *Starting Device Address is optional; if left blank, the control    │   Help   │
  unit number is used.                                     └──────────┘
```

6. Click the **Add New...** button on the right-hand side to create the control units and devices that will be connected to PROC1.21 CNC to represent the images available via PROC1.20 CTC.

```
Create CTC Control Units/Devices                                    [X]

    Create CTC Control Unit(s) to be connected to PROC1.21 S CNC SHR

            Number    Starting
            of        Device        Destination          ┌──────────┐
  Control   Devices   Address*      on PROC1             │┄┄┄OK┄┄┄┄│
  Unit                                                    └──────────┘
  [5110]    [4]       [    ]         LPAR11               ┌──────────┐
                                                          │  Cancel  │
  [5120]    [4]       [    ]         LPAR12               └──────────┘
                                                          ┌──────────┐
  *Starting Device Address is optional; if left blank, the control    │   Help   │
  unit number is used.                                     └──────────┘
```

7. Enter control unit numbers 5110 and 5120 to represent the two images and click **OK**. The **Edit CTC Connection** dialog will reappear as follows:

Edit CTC Connection

CTC connection routed via SWCH_1

Control units on PROC1.20 S CTC SHR

Control Unit	Devices	UA OS	Destination on PROC1
4110	4110,4	00	LPAR11
4120	4120,4	00	LPAR12

[Edit...] [CU<>Proc...] [I/O Devices...]
[Add New...] [Add Existing...] [Remove]

Control units on PROC1.21 S CNC SHR

Control Unit	Devices	UA OS	Destination on PROC1
5110	5110,4	00	LPAR11
5120	5120,4	00	LPAR12

[Edit...] [CU<>Proc...] [I/O Devices...]
[Add New...] [Add Existing...] [Remove]

[Cancel]   [Help]

This shows that the connection is fully specified with all images duly represented by control units. Clicking on **I/O Devices...** followed by **Dev <=> Proc...** shows that each of the devices has been set up with explicit candidate lists so that no partition has access to the devices that represent it.

8. Close the **Edit CTC Connection** dialog and choose **CTC connections...** from the **Edit** menu to open the **CTC Connections** dialog. The connection between PROC1.20 CTC and PROC1.21 CNC will appear in the dialog.

CTC Connections

CTC or FC CHPID	Type	Entry Port	Dynamic switch	Exit Port	CNC/FCV or FC CHPID	Type
PROC1.20	CTC	E0	01	E1	PROC1.21	CNC

[Edit...]
[Details...]
[Create...]
[Delete]
[Fiber Link...]
[Unused...]

[Close]
[Help]

Sort by
◉ CTC or FC (left)
◯ CNC/FCV or FC (right)
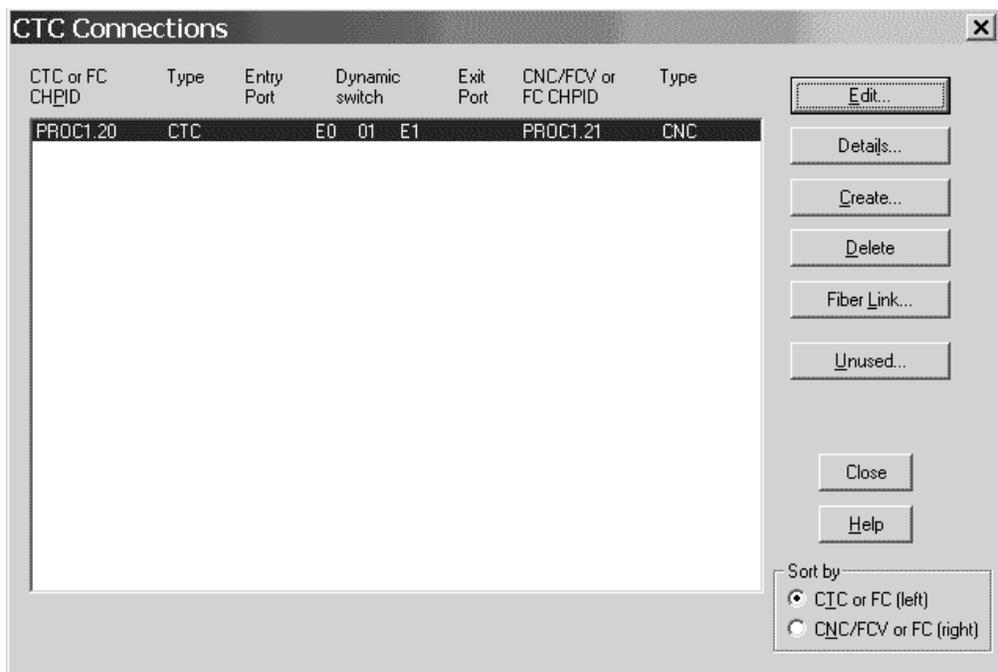
9. Select the entry in the list box and click **Details...** to open the **View CTC Connection Details** button.

View CTC Connection Details

	CTC or FC CHPID:		PROC1.20 S CTC SHR					
	Route:		SWCH_1(E0..E1)					
	CNC/FCV or FC CHPID:		PROC1.21 S CNC SHR					

Image on PROC1	Control Unit	Devices	OS	UA	Devices	OS	Control Unit	Image on PROC1
LPAR11	4110						5110	LPAR11
LPAR11	4120	4120,4		00	5110,4		5110	LPAR12
LPAR12	4110	4110,4		00	5120,4		5120	LPAR11
LPAR12	4120						5120	LPAR12

Cancel   Help
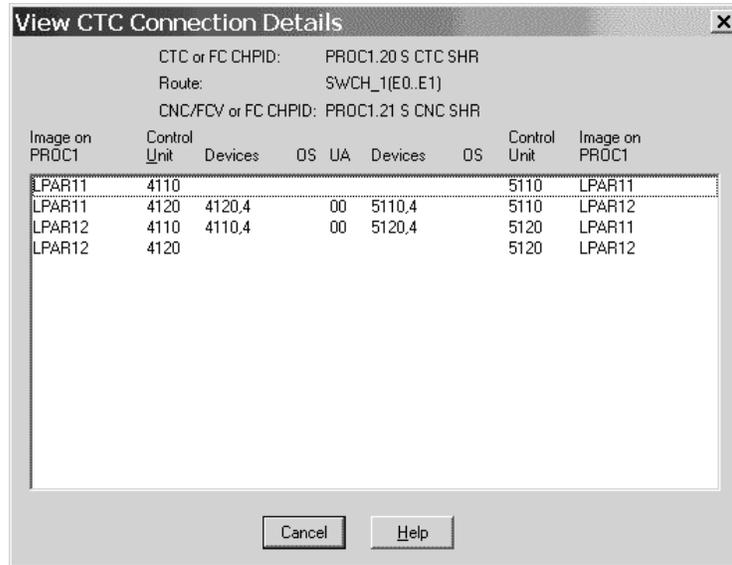
10. This gives another view of the connection, indicating that the two partitions are able to communicate with each other but not to themselves because of the setup of the device candidate lists.

11. Click **Cancel** to return to the **CTC Connections** dialog.

Create the second CTC connection

The second connection is between PROC2.20 CTC and PROC1.21 CNC. In this case, we have already created control units and devices that may be used to represent partitions LPAR11 and LPAR12 via PROC1.21 CNC; we need only connect them appropriately to PROC2.20 CTC. We will have to create a control unit and 4 devices to represent the single partition (LPAR21) that is available via PROC2.20 CTC.

1. Click **Create...** on the **CTC Connections** dialog.
2. Click the **CTC or FC CHPID** button and select PROC2.20 CTC DED.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC1.21 CNC SHR.
4. Click **OK**. The **Accept Proposed CTC Control Units** dialog to propose control units 4110 and 4120 to be connected to PROC2.20 CTC so as to represent LPAR11 and LPAR12, respectively.

Accept Proposed CTC Control Units

CTC connection routed via SWCH_1

Control units on PROC2.20 S CTC DED

Control Unit	Devices	UA OS	Destination on PROC1
4110	4110,4	00	LPAR11
4120	4120,4	00	LPAR12

Control units on PROC1.21 S CNC SHR

Control Unit	Devices	UA OS	Destination on PROC2
			LPAR21

The control units appearing in gray are those proposed by HCM to help complete this CTC connection. Accept or Reject these changes before continuing the Edit session.

[Accept]   [Reject]   [Help]

5. Click **Accept** to connect the control units to PROC2.20. The **Edit CTC Connection** dialog will appear showing that the CTC side has been defined. Note that the CNC side of the dialog shows that there is only one image to be represented. This is because PROC2.20 CTC is dedicated to one partition, LPAR21. As yet, there is no control unit to represent LPAR21.
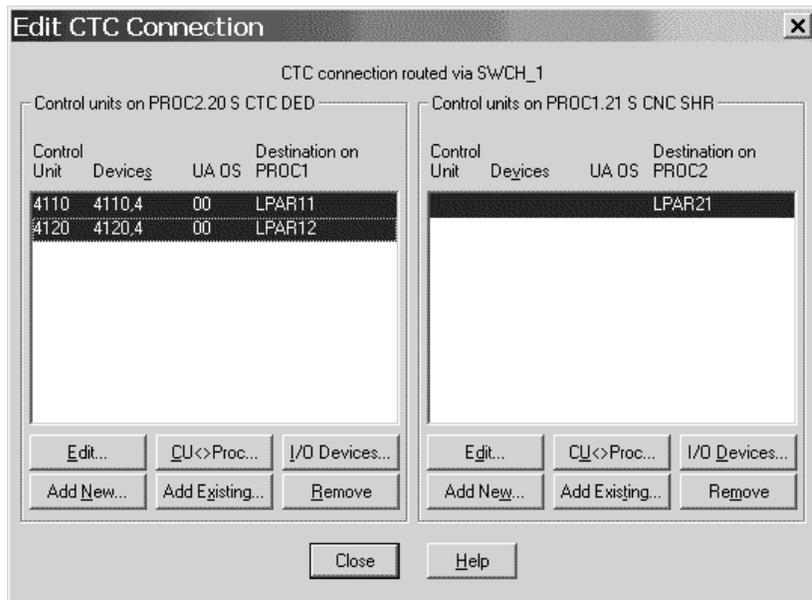
Edit CTC Connection

CTC connection routed via SWCH_1

Control units on PROC2.20 S CTC DED

Control Unit	Devices	UA OS	Destination on PROC1
4110	4110,4	00	LPAR11
4120	4120,4	00	LPAR12

[Edit...] [CU<>Proc...] [I/O Devices...]
[Add New...] [Add Existing...] [Remove]

Control units on PROC1.21 S CNC SHR

Control Unit	Devices	UA OS	Destination on PROC2
			LPAR21

[Edit...] [CU<>Proc...] [I/O Devices...]
[Add New...] [Add Existing...] [Remove]

[Close]   [Help]

6. Click the **Add New...** button on the right hand side to create the control unit and devices that will be connected to PROC1.21 CNC to represent the partition (LPAR21) available via PROC2.20 CTC. Use control unit 5210 and devices 5210,4.

7. Close the **Edit CTC Connection** dialog. The **CTC Connections** dialog will reappear showing the new connection.

Create the third CTC connection

The third connection is between PROC2.20 CTC and PROC3.21 CNC. In this case we have a control unit and 4 devices that may be used to represent LPAR21 via PROC2.20 CTC, but we will need to create a control unit and devices to represent PROC3 via PROC3.21 CNC.

1. Click **Create...** on the **CTC Connections** dialog.
2. Click the **CTC or FC CHPID** button and select PROC2.20 CTC DED.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC3.21 CNC DED.
4. Click **OK**. The **Accept Proposed CTC Control Units** dialog to propose control unit 5210 to be connected to PROC3.21 CNC so as to represent LPAR21.
5. Click **Accept** to connect the control unit to PROC3.21.
6. The **Edit CTC Connection** dialog will appear showing that the CNC side has been defined, but there is no control units to represent image PROC3 on PROC2.20 CTC.
7. Click the **Add New...** button on the left hand side to create the control unit and devices that will be connected to PROC2.20 CTC to represent PROC3 via PROC3.21 CNC. Use control unit 4300 and devices 4300,4.
8. Close the **Edit CTC Connection** dialog. The **CTC Connections** dialog will reappear showing the new connection.

Create the last CTC connection

The final connection is between PROC1.20 CTC and PROC3.21 CNC. In this case control units have already been created to represent all of the images involved. During the create process, HCM will propose these control units to be connected to the CHPIDs, and we need only accept these proposals to complete the connection.

1. Click **Create...** on the **CTC Connections** dialog.
2. Click the **CTC or FC CHPID** button and select PROC1.20 CTC DED.
3. Click the second **CNC/FCV or FC CHPID** button and select PROC3.21 CNC DED.
4. Click **OK**. The **Accept Proposed CTC Control Units** dialog to propose control unit 4300 to connected to PROC1.20 CTC so as to represent PROC3, and control units 5110 and 5120 to be connected to PROC3.21 CNC so as to represent LPAR11 and LPAR12.
5. Click **Accept** to connect the control units. The **Edit CTC Connection** dialog will appear showing the fully defined connection.
6. Close the **Edit CTC Connection** dialog. The **CTC Connections** dialog will reappear showing the new connection. Close the **CTC Connections** dialog.

Define the CTC devices to the OS configuration

Assuming one OS configuration (OS1) will be used for all four images in the connection, the last step is to define the new devices to the OS configuration.

1. Select **Edit:OS configurations...** to open the **Operating System Configurations** dialog.

Operating System Configurations

OS Configurations, EDTs, and Esoterics

OS1	MVS Configuration for all images

I/O Devices...

Generics...

Consoles...

Edit...

Create OS Config...

Create EDT...

Create Esoteric...

Delete

Cancel

Help
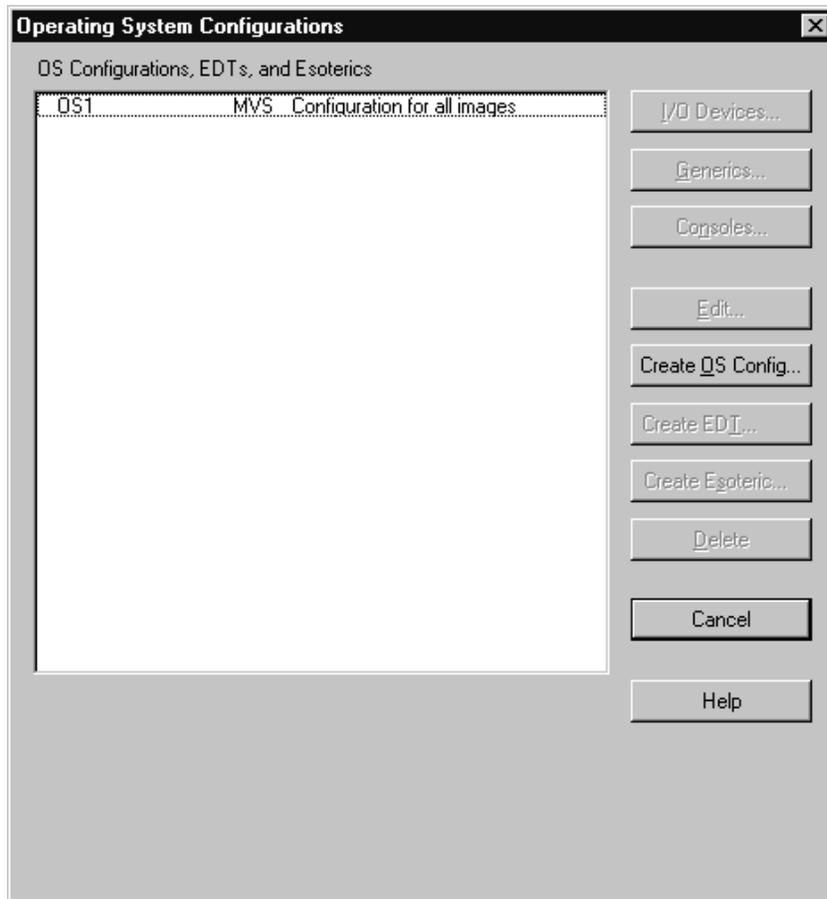
2. Select OS1 from the list box and click **I/O Devices...** to open the **List Devices for OS Configuration** dialog.

3. Click **Add...** to open the **Add I/O Devices to OS Configuration** dialog. This will show all the devices in the configuration that are not yet defined to OS1.

4. Toggle **Use Filter** to "on" and click **Filter...** to open the **Filter I/O Device List** dialog.

Filter I/O Device List

List only the devices that match all of the following criteria:

Type   SCTC ▼

Class   All ▼

Serial Number   [   ]

Description   [   ]

Control Units
- ◉ No CU Filtering
- ○ Not Connected
- ○ Connected

Specific CUs: [▼] or [▼] or [▼] or [▼]

Operating Systems
- ◉ No OS Filtering
- ○ Not Defined
- ○ Defined

Processor/Partition [▼]

[OK]
[Cancel]
[Help]
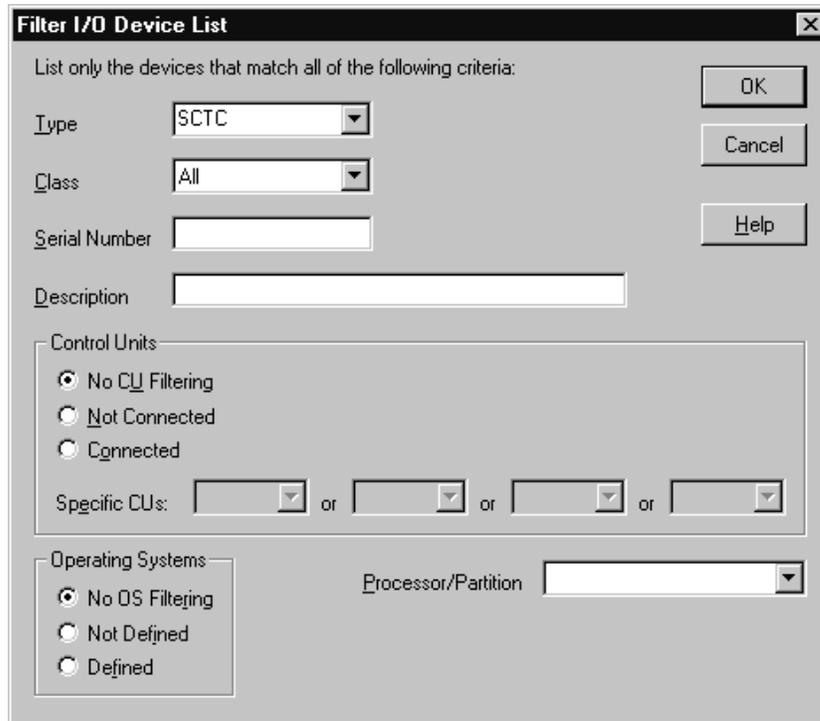
5. Enter "CTC" in the **Type** field and click **OK**. The **Add I/O Devices to OS Configuration** will reappear showing only CTC devices.

Add I/O Devices to OS Configuration

Add devices to OS Config   OS1

Device	Type	OS Control Units
+ 4110,4	SCTC	4110
+ 4120,4	SCTC	4120
+ 4300,4	SCTC	4300
+ 5110,4	SCTC	5110
+ 5120,4	SCTC	5120
+ 5210,4	SCTC	5210

[OK]
[Cancel]
[Help]

View
- ◉ Control Units   ○ Description   ○ Serial Number

☑ Use Filter

[Filter...]
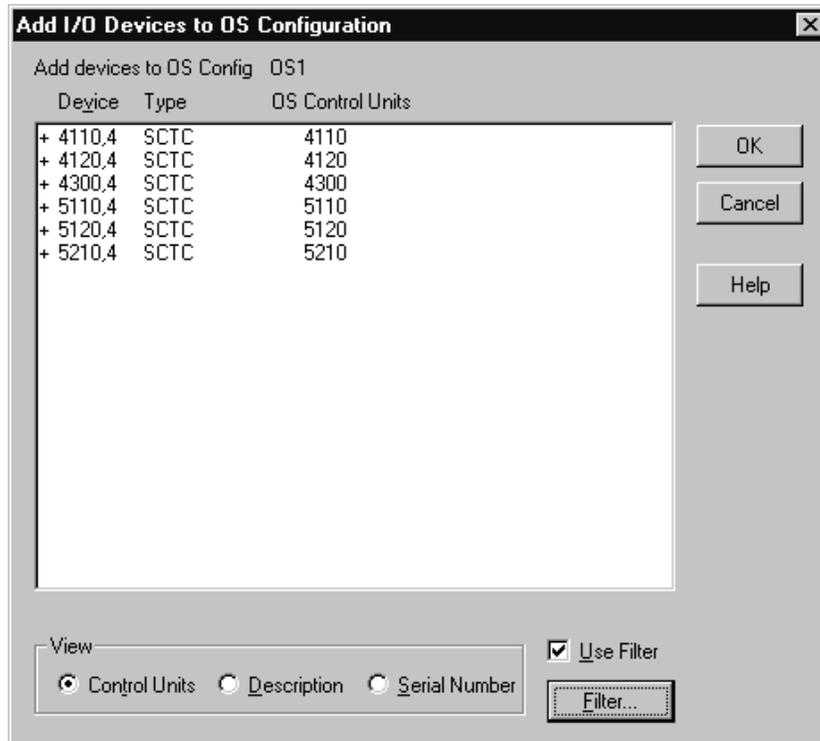
6. Select all of the devices in the list and click **OK**. The **Define Device <==> OS Configuration Parameters** dialog will appear.

Define Device <==> OS Configuration Parameters

					OK
Device Number:	4110		OS Config:	OS1	
Number of devices:	24		Type:	MVS	Cancel
Type:	SCTC		Description:	Configuration for all images	
					Help

Parameter	Value		Req	Description	
OFFLINE	☐			Device considered online or offline at IPL	?
DYNAMIC	☑			Device has been defined to be dynamic	?
?PARM15?					?

7. Set up the OFFLINE and DYNAMIC parameters as appropriate and click **OK**. This will define all of the CTC devices to the OS configuration. Close all open dialogs.

Scenario III – Defining a fiber cabling infrastructure

In this example, the logical configuration already exists. Within this scenario we will describe how you can add the physical objects to an already existing logical configuration, for example, after a **Load** process.

We will create two cabinets, a processor zone cabinet and a central cabinet which is presumably located near one or more switches. We will trunk the two cabinets together, then trunk the zone cabinet to a processor. Finally, we will route a switch-to-processor connection through the cabinets using a labeled fiber jumper.

Create the cabinets

We will create the zone cabinet first.

1. Select **Create:Cabinet** from the menu. This opens the **Create Cabinet** dialog.
2. Select "IBM Zone Cabinet" for Model and enter "P" for the cabinet ID to denote "processor zone". Select "Near Processors" for the Suggested Graphical Position of Patchport. You can also enter a cabinet description and a serial number.
3. Click **OK**. The new cabinet appears on the right side of the diagram.
4. Now create the central cabinet with "IBM Main Distribution Frame" for Model, ID "C", and suggested graphical position "Central".

Trunk the two cabinets together

We will now connect the two cabinets back-to-back using a permanent trunk.

1. Edit the processor zone cabinet by either selecting **Edit:Cabinet** and choosing the cabinet from the list or by double-clicking on the cabinet in the diagram. This opens the **Cabinet** dialog.
2. Add a panel by clicking "Add" in the panel section. This opens the **Add Panels** dialog. Enter the required data in this dialog to obtain the requested panels.
3. Click the "Front" radio button in the View section of the **Cabinet** dialog. Select 7 patchports and click "Connect". This opens the **Select Connection Target(s)** dialog.
4. In the list box, double-click on the back-marked central cabinet. This expands the list, showing the backs of all patchports in that cabinet.
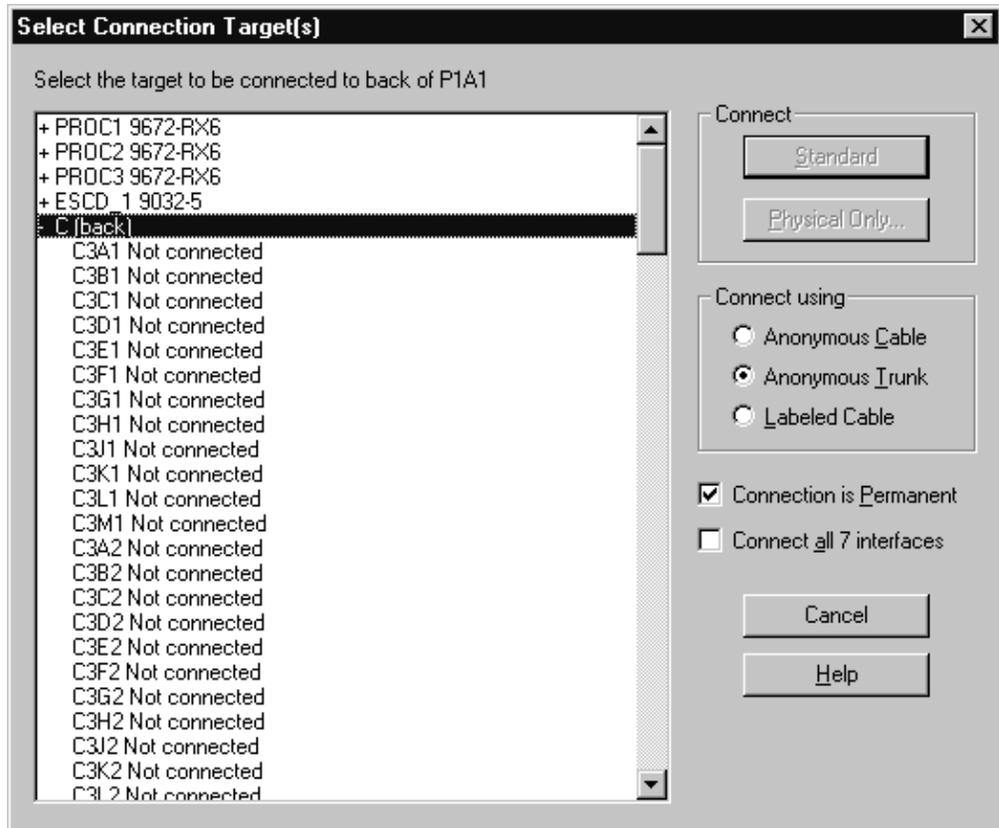
5. Select the first patchport, C3A1.
6. Check "Connect all 7 interfaces".
7. Select **Anonymous Trunk** and click **OK**.

    The first 7 patchports of the two cabinets have now been trunked together.

Trunk the zone cabinet to the processor

We will now connect the Processor Panel to some CHPIDs on the processor, using an anonymous (unlabeled) permanent trunk.

You should now be in the **Cabinet** dialog. Change the radio button from "Front" to "Back" in the View section and click Connect. This opens the **Select Connection Target(s)** dialog.

Select Connection Target(s)

Select the target to be connected to back of P1A1

```
+ PROC1 9672-RX6
- PROC2 9672-RX6
    PROC2.20 S CTC SHR Not connected
    PROC2.21 S CNC DED Not connected
    PROC2.22 S CNC DED Not connected
    PROC2.23 S CNC DED Not connected
    PROC2.24 S CNC DED Not connected
    PROC2.25 S CNC DED Not connected
    PROC2.26 S CNC DED Not connected
    PROC2.27 S CNC DED Not connected
    PROC2.28 S CNC DED Not connected
    PROC2.29 S CNC DED Not connected
    PROC2.2A S CNC DED Not connected
    PROC2.2B S CNC DED Not connected
    PROC2.2C S CNC DED Not connected
    PROC2.2D S CNC DED Not connected
    PROC2.2E S CNC DED Not connected
    PROC2.2F S CNC DED Not connected
    PROC2.30 S CNC DED Not connected
- PROC3 9672-RX6
    PROC3.20 S CNC DED Not connected
    PROC3.21 S CNC DED Not connected
    PROC3.22 S CNC DED Not connected
    PROC3.23 S CNC DED Not connected
    PROC3.24 S CNC DED Not connected
    PROC3.25 S CNC DED Not connected
    PROC3.26 S CNC DED Not connected
    PROC3.27 S CNC DED Not connected
```

Connect
- Standard
- Physical Only...

Connect using
- ○ Anonymous Cable
- ● Anonymous Trunk
- ○ Labeled Cable

☐ Connection is Permanent
☑ Connect all 7 interfaces

Cancel

Help

1. In the list box, double-click on the processor you are connecting. This expands the list, showing all CHPIDs in the processor.
2. Find a sequence of 7 free CHPIDs and select the first one.
3. Click "Connect all 7 interfaces".
4. Select **Labeled Cable** and click **OK**. The **Assign Cable** dialog will be opened six times in sequence for you to assign labels to the individual lines of the trunk.

Assign Cable

Connection: PROC2.20 --- P1A1

☐ Connection is Permanent

OK

ID: TR0001/00   Type: Multi Mode Jumper ▼   db Loss A: N/A   Cancel

Clear   Cable ID's...   Length: 60   db Loss B: N/A   Help

Comment: Patchport - CHPID Connection

Columns...

Unused cables:

ID	Type	Length	Last use	Time of last use	Comment	Loss A	Loss B
mmf1015	MMJ	60	PROC2.29 --- SWCH_1.EA	2002-06-20 09:18:18	PROC2.29 to SWCH1.EA	0.000	0.000
mmf1016	MMJ	60	PROC2.27 --- SWCH_1.E7	2002-06-20 09:19:15	PROC2.27 to SWCH1.E7	0.000	0.000

5. Label each line with the same trunk designator followed by a "/" and the line number. E.g. TR0001/00", "TR0001/01", etc.

6. Click **OK** to return to the **Cabinet** dialog, where you can see that the patchports are now connected to the CHPIDs, as well as to the patchports in the other cabinet.

Cabinet

ID: P   Description: IBM Zone Cabinet, Cabinet 1   Info...
Model: IBM Zone Cabinet   Edit...
Serial Number: 0987654321   Suggested Graphical   Near processors
Position of Patchports:

Patchports for panel 1 "P1"

View: ○ Front ● Back   Panel size is 10×10   Patchports are sequenced row first

						Panels
⊔ PROC2.20 B	⊔ PROC2.21 B	⊔ PROC2.22 B	⊔ PROC2.23 B	⊔ PROC2.24 B	⊔ PROC2.25 B	1 2
⊓ P1A1	⊓ P1B1	⊓ P1C1	⊓ P1D1	⊓ P1E1	⊓ P1F1	
⊓ C3A1	⊓ C3B1	⊓ C3C1	⊓ C3D1	⊓ C3E1	⊓ C3F1	

Attributes...
Add...
Remove
Move Up
Move Down

B	B	B	B	B	B
⊓ P1A2	⊓ P1B2	⊓ P1C2	⊓ P1D2	⊓ P1E2	⊓ P1F2
⊓ P1A3	⊓ P1B3	⊓ P1C3	⊓ P1D3	⊓ P1E3	⊓ P1F3
⊓ P1A4	⊓ P1B4	⊓ P1C4	⊓ P1D4	⊓ P1E4	⊓ P1F4
⊓ P1A5	⊓ P1B5	⊓ P1C5	⊓ P1D5	⊓ P1E5	⊓ P1F5

Patchport detail
☑ Show Front
☑ Show Entire Link
☐ Show Cables

Connect...   OK
Disconnect   Close
Attributes...   Help
ESCON Link...
Locate

7. Click **OK** on all open dialogs to close them.

Connect a switch to CHPIDs through the patchport cabinets

Permanent connections between pairs of patchports must be made before permanent connections between patchports and CHPIDs. It is also necessary to make all permanent (trunk) connections among patchports before adding those patchports to fiber links.

We will connect an existing switch port to one of the trunked CHPIDs. Although the CHPIDs are already connected to patchports in the cabinet, we will specify the CHPID as the connection target. The patchports will automatically be connected between the switch port and the CHPID.

1. Edit the switch by either selecting **Edit:Switch** and choosing the switch from the selection list or by double-clicking on the switch in the diagram. This opens the **Switch** dialog.
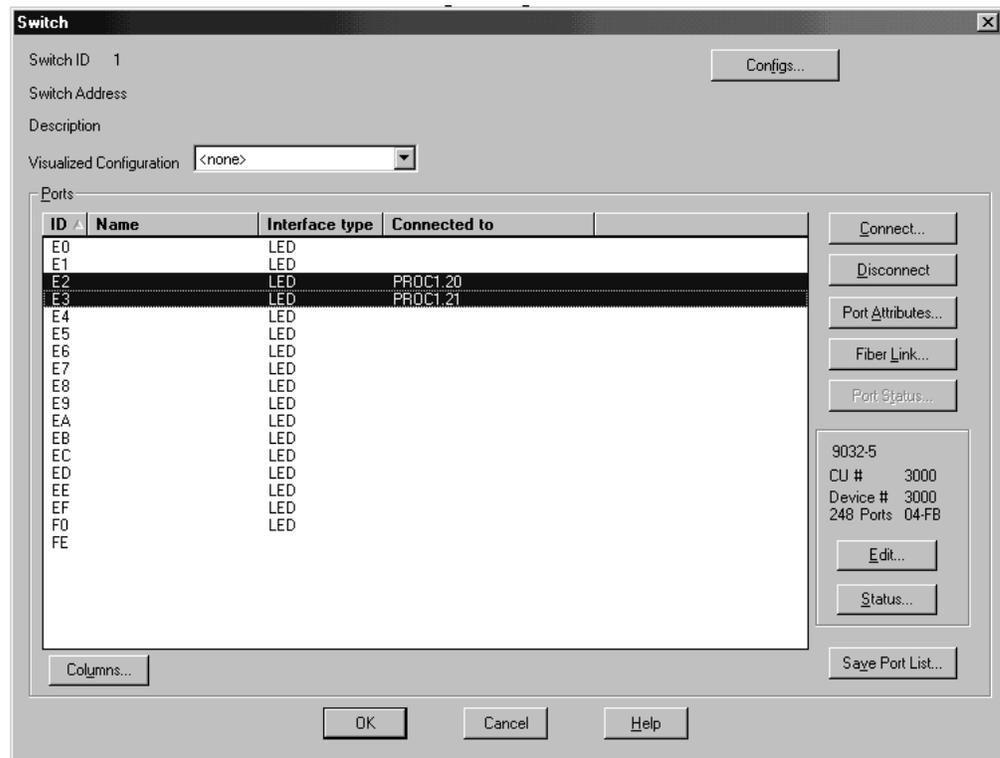
2. Select one of the free switch ports and click **Connect**. This opens the **Select Connection Target(s)** dialog.
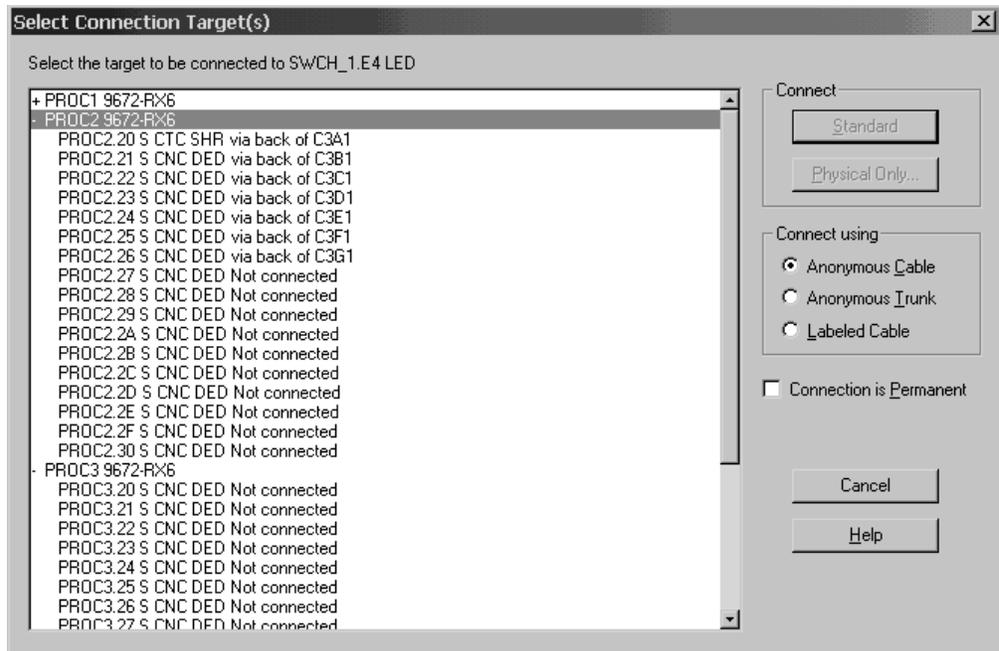
3. In the list box, double-click on the processor with the trunked CHPIDs. This expands the list, showing all CHPIDs in the processor.

4. Double-click on one of the trunked CHPIDs. The connection is made between the switch port and the CHPID.

Assign a cable to the new connection

1. In the **Switch** dialog, select the newly connected switch port and click **Fiber Link**. This opens the **Edit Fiber Link** dialog which shows the composition of the fiber link in the box to the right.

2. Select the "---" between the switch port and patchport C3A1 and click **Edit Cable**. This opens the **Assign Cable** dialog.

3. Fill in the cable's ID, type, and length, then click **OK**. The cable's new label now appears in the **Fiber link** dialog.
4. Click **OK** on all open dialogs to close them. The new connection between the switch and the CHPID through both patchports can be seen on the diagram.

Connect a controller to the CHPID via the switch

1. Edit an existing controller by either selecting **Edit:Controller** and choosing the controller from the list or by double-clicking on the controller in the diagram. This opens the **Select controller...** dialog. Select a controller and click **OK**. This opens the **Controller** dialog.
2. Select a free channel interface to connect and click **Connect**. This opens the **Select Connection Target(s)** dialog.

Select Connection Target(s)

Select the target to be connected to 3990 #1.SC0.A[00] S

```
+ PROC1 9672-RX6
+ PROC2 9672-RX6
+ PROC3 9672-RX6
- SWCH_1 9032-5
    SWCH_1.E0 LED from PROC1.20
    SWCH_1.E1 LED from PROC1.21
    SWCH_1.E2 LED from PROC2.28
    SWCH_1.E3 LED from PROC3.21
    SWCH_1.E4 LED from PROC2.20
    SWCH_1.E5 LED Not connected
    SWCH_1.E6 LED Not connected
    SWCH_1.E7 LED Not connected
    SWCH_1.E8 LED Not connected
    SWCH_1.E9 LED Not connected
    SWCH_1.EA LED Not connected
    SWCH_1.EB LED Not connected
    SWCH_1.EC LED Not connected
    SWCH_1.ED LED Not connected
    SWCH_1.EE LED Not connected
    SWCH_1.EF LED Not connected
    SWCH_1.F0 LED Not connected
+ 3990 #1
```

Connect
- Standard
- Physical Only...

Connect using
- ● Anonymous Cable
- ○ Anonymous Trunk
- ○ Labeled Cable

☐ Connection is Permanent

Cancel

Help

3. Choose the switch that we trunked to a CHPID from the list and double-click it to expand it. This lists all of the installed ports in the new switch.

4. Select a port with no connection and click **Standard** in the **Connect** box.

5. Since we are connecting through a switch, the **Processor and CHPID Connections** dialog is opened for you to choose the CHPIDs to which the control unit will be defined.

Processor and CHPID Connections

Tabs: PROC1 | PROC2 | **PROC3** | Summary

PROC3 CHPIDs connected to control unit 0C0C via 3990 #1.SC0.A

Reachable CHPIDs

CHPID	Type	Mode	Port	Dynamic director	Device count

Add >

< Remove

Show
- ☐ Partitions
- ☐ Control units

Sort
- ● By CHPID
- ○ By Device count

Connected CHPIDs

CHPID	Type	Mode	Port	Dynamic director	DLA	Device count
21	CNC	DED	01.E3	01	E5	0

No. of Managed CHPIDs: 0

Attributes...

OK   Cancel   Help

6. Select the CHPID that we trunked to this switch and click **Add>>** to move this CHPID to the **Connected CHPIDs** list.

7. Click **OK**. The connection is now made and we are returned to the **Controller** dialog.

8. Click **OK** on all open dialogs to close them. The connection between the controller and the switch appears on the diagram. When the controller is selected, the full connection is highlighted from the controller through the switch and the patchports, to the CHPID.

9. Save your configuration file.

Scenario IV – Configuration changes for installing a FICON bridge card in a 9032–5 ESCON director

The new FICON Conversion Channels (FCVs), with their wider bandwidth compared to classical ESCON channels, allow connecting more I/O equipment to a single channel. To use the FCV channels, they must be connected to a FICON bridge card installed in a 9032-5 ESCON director. To install FCV channels and a FICON bridge card in an 9032-5 you must follow some configuration rules:

- FCV channels can only be installed in specific locations in the processor and have particular CHPID numbers depending on the processor type.
- FICON bridge cards can have only one port address (and link address), on a modulo 8 + 4 port address.
- The 7 consecutive subsequent port addresses of a FICON bridge port address are used by the FICON bridge card and cannot be used for other connections.

The FICON conversion utility can be used for adapting an IODF to use FCV channels. The following scenario shows an example of how an I/O configuration can easily be changed using the utility. The starting point in this scenario is a configuration in which a CNC CHPID D4 is connected to a port 74. CNC CHPID D7 is also connected to the ESCD. Three additional CHPIDs are connected to ports 75, 78, and 79. The objective of the configuration changes is to have a FICON bridge card at position 74, and CHPID D4 must be converted from CHPID type CNC to FCV. Also, all attached I/O equipment of CHPID D7 will be disconnected and reconnected to the FCV CHPID D4 (aggregation of CHPID D7 to D4) to make use of the wider bandwidth of the FCV channel. The original connection between ESCD port E0 and CHPID D7 will be disconnected (since this connection is no longer used). The connections between ports 75, 78, and 79 and their channels must be relocated to different positions on the director, since the original port addresses can no longer be used after installing a FICON Bridge card. Relocating the connections from 75, 78, and 79 to different positions using this utility will not affect the connectivity of the respective CHPIDs to their I/O equipment.

To perform the changes, proceed as follows:

1. Invoke the **Convert ESCON Port to FC...** utility from the **Utilities** menu.
2. The wizard will appear and explain the following steps.

Convert ESCON Port to FC Port

This utility helps you install a Fibre Channel port card on an ESCON director and rearrange ESCON ports to accommodate it.

In order to use this utility, you must have at least one FC-eligible port connected to an FCV-eligible CHPID. You may need to use the "Move Port Connections" wizard to ensure that your target port is connected to an appropriate CHPID.

The process has up to five steps:

1. Choose an ESCON port to be converted to an FC port.

2. Optionally choose CHPIDs on the same director to be aggregated to the CHPID connected to the target port.

3. Optionally choose the aggregated CHPIDs that are to be disconnected from the target director.

4. Pick destination ports for any connections displaced by the FC port card.

5. Examine a summary of the changes that are to be made, and click "Finish" to execute the operation.

No updates are performed until the Finish button is clicked in the last step. It is safe to preview the entire process as a dry run.

Click Next to begin.

[< Back] [Next >] [Cancel] [Help]

Click the **Next >** button to continue.

3. A list of all possible ports that can be used to hold a FICON bridge card will be shown. The only selectable ports will be those with an appropriate port address (starting with 04 modulo 8 + 4 of 9032-5) and which are already connected to an FCV or to an FCV-eligible CHPID (e.g., a CHPID whose type can be changed to FCV). All ports that are not usable for a FICON bridge card are grayed out.

Select port 74 of director 05 connected to CHPID D4 of processor SYSB and

---

Convert ESCON Port to FC Port

Choose the ESCON port group that you want to convert to an FC port.

Ports	CHPID to be converted to FCV
05.(04-0B)	
05.(0C-13)	
05.(14-1B)	
05.(1C-23)	
05.(24-2B)	
05.(2C-33)	
05.(34-3B)	
05.(3C-43)	
05.(44-4B)	
05.(4C-53)	
05.(54 5B)	
05.(5C-63)	
05.(64-6B)	
05.(6C-73)	
05.(74-7B)	SYSB.D4 S CNC SHR
05.(7C-83)	
05.(84-8B)	
05.(8C-93)	
05.(94-9B)	
05.(9C-A3)	
05.(A4-AB)	
05.(AC-B3)	CCFA.00 S CNC DED
05.(B4-BB)	

[< Back] [Next >] [Cancel] [Help]

---
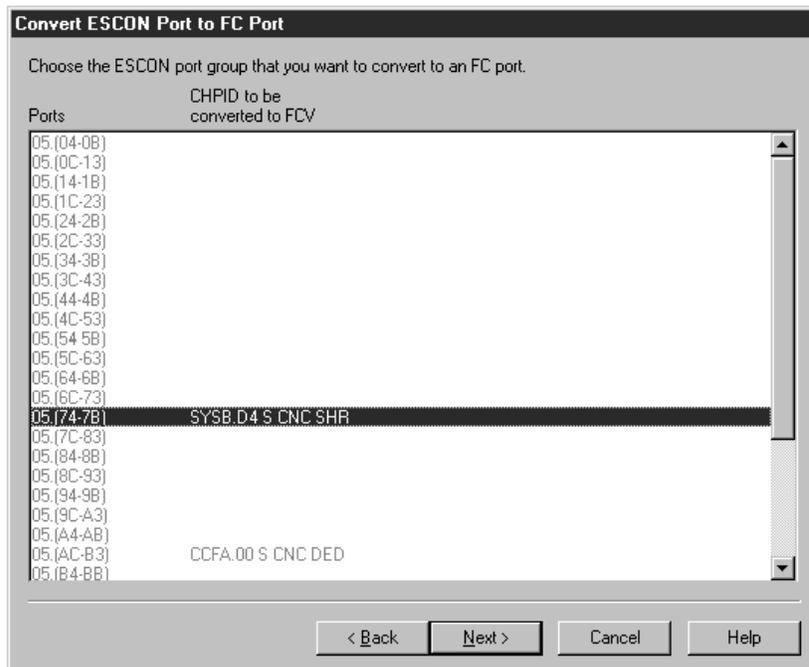
click **Next >**.

4. In the following step, you have the chance to aggregate I/O equipment of other CHPIDs to CHPID D4. Select CHPID D7 of processor SYSB to be aggregated to D4 and click the **ADD >** button.
In the bottom part of the window you will see CHPIDs and their I/O
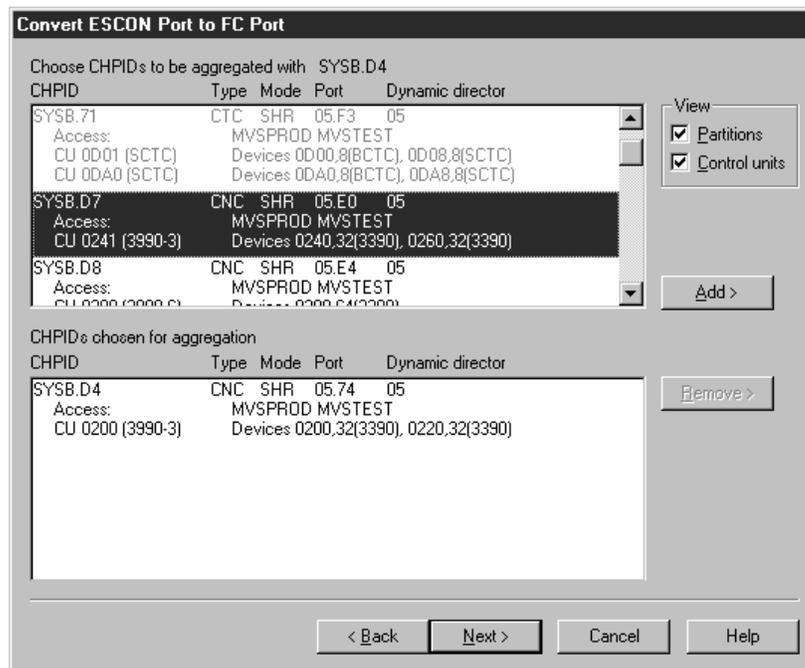
equipment as it will be aggregated. Click **Next >** to continue.

5. In the following window, you can specify whether or not to disconnect the just selected CHPID for aggregation of D7 from its port of the ESCD.
Click **Next >** to continue to displace port connections.
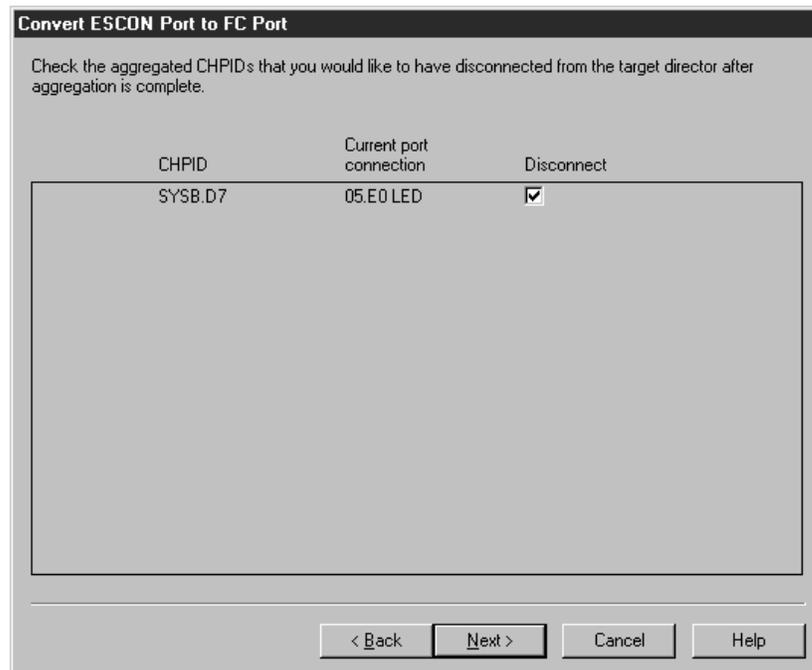
6. You can now determine where to move port connections from connections to port addresses which must be freed, since they follow directly after the port

address of the FICON bridge port address. For port address 75 connected to CHPID DF of processor SYSB, specify the new port address 06. For port address 78 connected to CHPID EA of processor SYSB, specify the new port address 07. For 79 connected to CHPID DD of processor SYSB, specify the new port address 08. You can use the prompting capability to check for free port addresses. After selecting new port addresses for connections that must be displaced, click **Next >** to see the final summary.

7. You will now see the final summary of all changes to the configuration. This summary lists all affected objects including the changes to be performed to the IODF. Scroll down to examine the complete list, which will contain information such as:

   • CHPID that will be the FCV
   • Aggregated CHPIDs
   • Possibly disconnected CHPIDs
   • Control unit connections moved
   • Port connections moved
   • Uninstalled ports

   Scroll down to see the complete summary. You can save the summary to a file

```
Convert ESCON Port to FC Port

CHPID SYSB.D4's type will be changed from CNC to FCV.

The following CHPIDs will be aggregated with SYSB.D4:

  SYSB.D7

The following CHPIDs will be disconnected:

  CHPID      Port
  SYSB.D7    05.E0

Port names will be moved with their ports.

The following connections will be moved:

                                 From     To
SYSB.DF                          05.75    05.06
SYSB.E4                          05.78    05.07
SYSB.DD                          05.79    05.08

Click Finish to accept these changes.                    Save this list...


              < Back     Finish      Cancel      Help
```

   to store the list of changes for archive purposes or to use the list as a worksheet for guidance for changing the cable connections directly at the hardware box. As long as you do not click the **Finish** button, no changes are written to the HCM configuration and no changes are made to the IODF. It is still possible to refrain from performing the specified changes. Click the **Finish** button to allow the wizard to complete. Confirm the configuration changes after you have examined all the listed items in the summary and agree with it.

If you save this summary to a file, you will have the following list of configuration changes:

• CHPID SYSB.D4's type will be changed from CNC to FCV.
• The following CHPIDs will be aggregated with SYSB.D4:

```
        SYSB.D7
```
- The following CHPIDs will be disconnected:

```
  CHPID      Port
  SYSB.D7    05.E0
```
- Port names will be moved with their ports.
- The following connections will be moved:

```
                              From   To
  SYSB.DF                     05.75  05.06
  SYSB.E4                     05.78  05.07
  SYSB.DD                     05.79  05.08
```
- The IODF will be updated as follows:

```
  CHPID Type changes          From   To
  SYSB.D4                      CNC    FCV
```
- The following control unit connections will be moved:

```
                              From       To
  CU 0241 (3990-3)            SYSB.D7    SYSB.D4
```
- Any devices using a source CHPID as the preferred CHPID will be updated to use the target CHPID.
- The following port connections will be moved:

```
                              From   To
  SYSB.D7                     05.E0  -----
  SYSB.DD                     05.79  05.08
  SYSB.DF                     05.75  05.06
  SYSB.E4                     05.78  05.07
```
- The switch configurations on director 05 (if any) will be updated as required.
- The following ports will be uninstalled:

```
  05.75
  05.76
  05.77
  05.78
  05.79
  05.7A
  05.7B
```

Scenario V – Establish a PPRC connection consisting of several PPRC links

PPRC (peer-to-peer remote copy) connections are direct connections between DASD controller subsystems that are used primarily to provide a hot standby capability. These connections can be point-to-point from one DASD controller to another, or they may pass through switches, just as connections from CHPIDs to control units can. Note that HCM provides methods to physically document PPRC connections, and to export PPRC connection information into CESTPATH command and GEOPARM control statement format for further use on the host. HCM does not validate whether the entered PPRC links are valid links in a real configuration, but HCM only provides documentation for this type of connections. In this scenario, we will set up a PPRC connection that consists of several single PPRC links. To demonstrate what you can do with this utility, some of the PPRC links will directly connect two subsystems with each other, while other PPRC links involve a director. Let us consider a basic configuration where we have the following I/O objects already defined:

- A FICON director, type 2032, Id=05

- An IBM Enterprise Storage Server® (ESS) DASD subsystem #1, consisting of control units 3000 and 3010, both having SSID 3000 specified, that has connections as follows:
  - Interface 04.A[0C] connected to director 05.40
  - Interface 04.B[0D] connected to director 05.41
- A second IBM Enterprise Storage Server (ESS) DASD subsystem #2, consisting of control units 4000 and 4010, both having SSID 4000 specified, that has connections as follows:
  - Interface 04.A[0C] connected to director 05.50
  - Interface 04.B[0D] connected to director 05.51
- Devices of type 3390 which are connected to the control units as follows: device range 3000,32 connected to control units 3000 and 3010, and device range 4000,32 connected to control units 4000 and 4010

To establish a PPRC connection, proceed as follows:

1. Edit the subsystem #1 by either selecting **Edit:Controller** and choosing the controller from the list or by double-clicking on the controller in the diagram. This opens the **Controller** dialog.
2. Select interface 12.A[8C] and click **Connect**. This opens the **Select Connection Target(s)** dialog.
3. Choose interface 12.A[8C] of DASD subsystem #2 as target, and click **Physical only** to connect both interfaces.
4. Repeat steps 2 and 3, now connecting the interfaces 12.B[8D] of both DASD subsystems #1 and #2 with each other. Click **Close** to leave the *Controller* dialog.
5. Select *Create: PPRC Connection*. This opens the **Create PPRC Connection** dialog. This utility will guide you through 4 pages on which you can specify certain information needed to create a PPRC connection.
6. Click on the first entry of the list (SSID 3000) in order to select the primary (source) subsystem of the PPRC connection, and click **Next**. This will show page two of the **Create PPRC Connection** dialog.
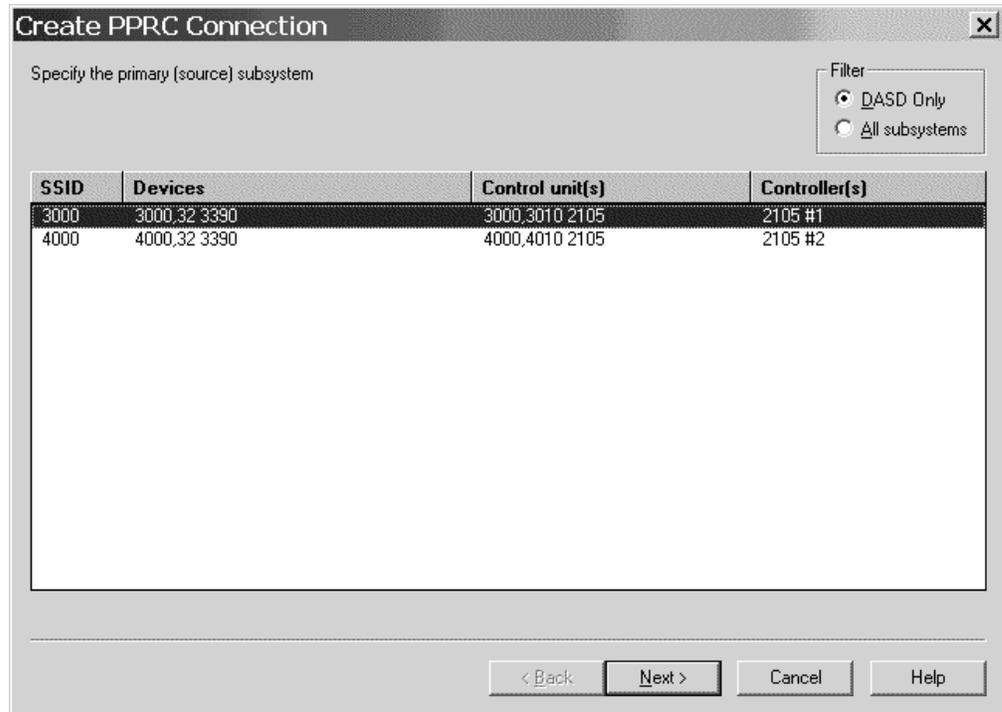
Figure 146. Create PPRC Connection (1)

7. Click on the second entry of the list (SSID 4000) in order to select the secondary (target) subsystem of the PPRC connection, and click again **Next** which will lead you to page 3 of the utility.

8. We want to create a PPRC connection consisting of four single PPRC links. Therefore, click the **More** button three times to obtain altogether 4 possible entries for a source and a target interface. As source interfaces on the left side of the dialog, select the interfaces 4.A[0C], 4.B[0D], 12.A[8C], 12.B[8D] which are interfaces of the DASD subsystem #1. As the latter two interfaces are directly connected to interfaces of DASD subsystem #2, HCM will automatically fill the entries on the right side (target interfaces). Your **Create PPRC Connection** dialog should now look similar to the following:
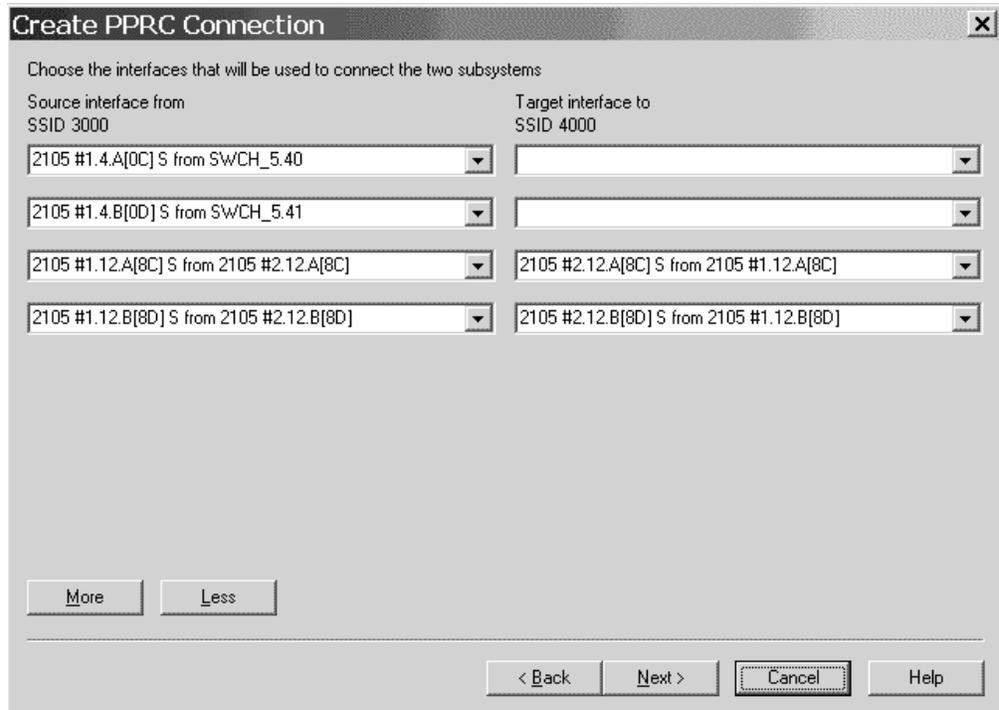
Figure 147. Create PPRC Connection (2)

Select the target interfaces 4.A[0C] and 4.B[0D] of DASD subsystem #2 to finish specifying the interfaces for both the source and the target subsystem.
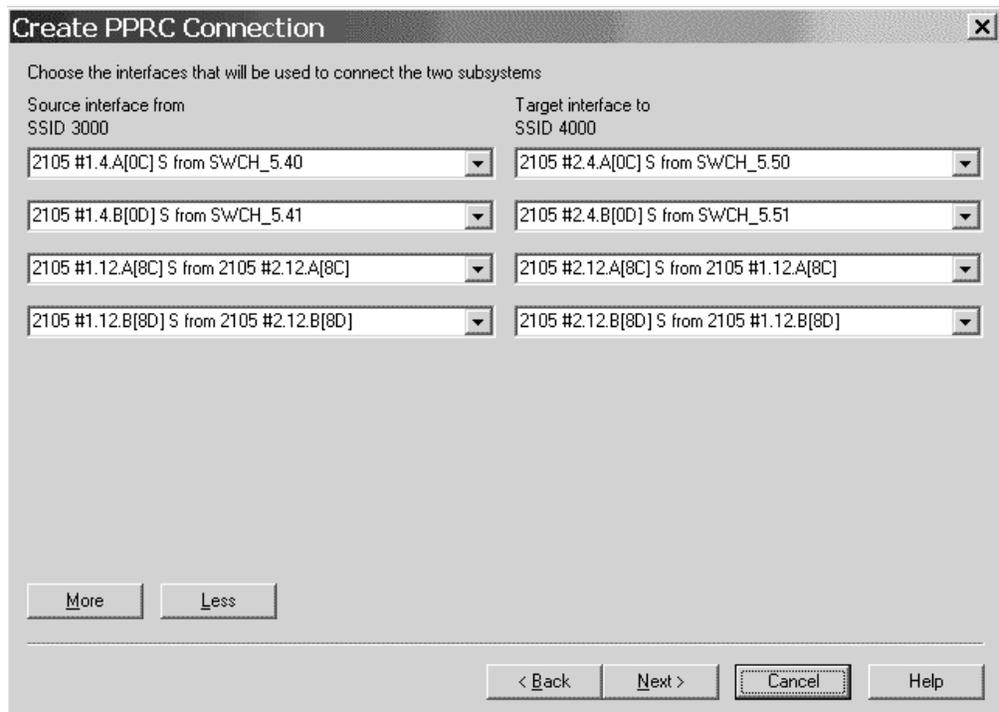
Figure 148. Create PPRC Connection (3)

Click **Next** which will show the last page of the utility.

9. Specify the parameters as needed, and finally click **Finish** to complete setting up the PPRC connection.

Figure 149. Create PPRC Connection (4)

You have now defined a new PPRC connection consisting of 4 single PPRC links. Using **Locate:PPRC Connection**, you can locate and highlight the PPRC connection in the diagram at any time.

Chapter 14. HCD-only tasks

The following tasks cannot be carried out using HCM. They must be done using HCD on the host.

Defining and viewing configuration data

You can use HCD to define and view configuration data for the following tasks.

Adding similar objects to a configuration

You can use the **Add like** function of HCD to add a new object, that has a definition similar to a previously added object. For devices and control units, this function copies also the defined relations.

Note: This **Add like** function is not available for z/VM HCD.

Copying configurations and merging IODFs

The existing **Repeat** function in HCD allows you to copy a selected object together with its dependent objects and I/O attachments not only within the same IODF but also to another IODF. This is useful if you want to consolidate configuration data from different IODFs in one IODF.

Note: For z/VM HCD, this function is available via REXX utility **CBDSMIGR**, see *z/VM: I/O Configuration* (SC24-6044).

Transferring partition configurations

Use the **Transfer (move) partition configs** function to transfer control units and devices attached to a channel path from one partition to another within the same IODF. In contrast to the **Repeat** function, you do not copy the partition and channel paths, but move the attached control units and devices to another partition, possibly in another processor.

Note: For z/VM HCD, this function is available via REXX utility **CBDSMIGR**, see *z/VM: I/O Configuration* (SC24-6044).

Migrating existing input data sets

Use HCD to migrate existing configuration data that was defined in IOCP, MVSCP, and HCPRIO input data sets to an IODF or update the IODF via control statements.

IODF import/export

You have to use HCD to send an IODF, and optionally its activity log file, to another (local or remote) system. You can export and import IODFs between different HCD releases.

Upgrade an IODF from an earlier version of HCD

When you want to use an MVS/ESA™ 4.2 or 4.3 IODF with a higher HCD release, you first have to upgrade the IODF into the new format before using it.

Comparing IODFs

You can use the HCD *Compare IODFs* function to compare two IODFs and report the differences between them. You can restrict the reports to certain views of the IODF:

- The processor view shows differences in the properties of partitions, channel paths, control units, and devices.
- The switch view shows differences in the properties of switches and switch configurations.
- The operating system view shows differences in device parameters, in features, in EDTs, in esoterics, in generics defined for EDTs, and consoles.

Printing reports

You can generate the following configuration reports about data in an IODF:

I/O Path Report
Shows the physically sensed I/O paths (with physical types) of the active system compared with the logical definitions of the paths (also the object types) of a specific IODF.

Note: This report is not available for z/VM HCD.

Supported Hardware Report
Contains information about the processors, control units and devices supported in your installation. This report is generated directly from the processor support modules and UIMs and reflects the latest support levels installed.

I/O Definition Reference
Contains a description of the parameters to define the device to the Channel Subsystem, and a description of the parameters and features to define the device to the MVS or VM operating system. This report is generated directly from the UIMs and reflects the latest UIM levels installed.

Querying supported hardware and installed UIMs

You can query supported processors, switches, control units, devices, and installed UIMs.

Note: With z/VM HCD, this information is only accessible via the *Supported Hardware Report*.

Viewing and printing the activity log

Use HCD to either view or print the activity log. Select *Print or compare configuration data* from the HCD primary task selection panel and then either the *View the activity log* or *Print the activity log* option.

Note: With z/VM HCD, you can view the activity log for an IODF using an editor or browser.

Support of large IODFs

Within large installations IODFs describing the entire I/O configuration may become too big for an effective management. To remedy this problem, while preserving the benefits of keeping a single repository for all I/O definitions, HCD can build subset IODFs based on configuration packages from a centrally administered master IODF. Configuration packages define subsets of the master

IODF, which allows selective distribution of I/O data to target systems, while keeping the processor token in sync. Configuration packages also allow configuration changes made at a remote system to be merged back into the master IODF.

Note: This support of large IODFs via a dialog is not available with z/VM HCD.

Activating and processing configuration data

You can use HCD to activate and process configuration data for the following tasks.

Building IOCDSs

You have to use HCD to build an IOCDS from a production IODF. Processors may have varying numbers of IOCDSs. One IOCDS is used at POR time to make the configuration data known to the channel subsystem.

HCD allows you to build an IOCDS for processors that do not have an SNA address defined as well as for processors with an SNA address configured in an S/390 microprocessor cluster and controlled by the Hardware Management Console. You can do the following:
- Display the IOCDS status
- Write a new IOCDS
- Switch IOCDS to become the active one for the next POR
- Enable/disable the write protection of the IOCDS (only for S/390 microprocessor cluster)

Managing IPL attributes

With HCD you can get a list of all CPCs that are configured in an S/390 microprocessor cluster and controlled by the Hardware Management Console. You can use this list to do the following:
- Display the IPLADDR and IPLPARM attribute values of the last and for the next IPL
- Modify IPLADDR and IPLPARM attribute parameters to be used for next IPL

Dynamic activate

HCD allows you to activate your configuration dynamically, that is to change a currently active configuration without the need to POR or IPL your system again.

You can do this for local processors HCD is running on as well as for systems that are configured in a sysplex. You can:
- Activate software configuration changes only
- Activate software and hardware configuration changes
- Activate a configuration sysplex wide
- Switch IOCDS for the next POR

Build configuration data sets

With HCD you can build configuration data sets for IOCP, JES3 initialization stream checker, and HCPRIO. You can also generate control statements for any OS configuration and switch configuration defined.

Building and processing CONFIGxx members

HCD allows to build individual CONFIGxx members from a production IODF. Existing CONFIGxx members can be updated or replaced.

For systems which are members of a sysplex, CONFIGxx members can be verified against selected systems. Responses are displayed in a message list.

Chapter 15. Importing and exporting data

The HCM Import and Export facility allows you to exchange data between your non-HCM processes, applications and data bases and the configuration data of HCM. The import and export data format used by HCM is designed to be compatible with relational databases. It is conceptually organized as tables, and the data exchange is done via data stored in tables as text files.

For more information on the data format, see "Data format for importing and exporting." For a list of tables that are available in HCM, see "Table summary" on page 333.

With the **Import Data** function you can use existing physical configuration data of your non-HCM application to update existing configuration data in HCM as well as to setup a new configuration file for HCM. Imported data overwrites existing HCM configuration data. In addition, import provides a method to build a configuration from the combination of a text file and an IODF.

The import function performs the following types of actions:
- Creating physical objects
- If necessary, resynchronizing the configuration file and the IODF
- Updating descriptive fields in both the configuration file and the IODF
- Creating or changing physical and some logical connectivity

With the **Export Data** function you can use configuration data of the HCM work and production files for your non-HCM applications and databases (for example, asset management). HCM configuration data can be stored in your external database for example, to create reports tailored to your needs. Or, you can create a list of devices including its VOLSERs.

Notes:

1. Not all IODF data is exported.
2. Not all data which is exported can be imported to update the configuration.
3. When you are exporting data from HCM, the export file does not contain the complete IODF information. For example, no logical data for control unit - processor connection. However, all physical objects and physical connections are contained in the export file.

> **Tip**
>
> For ease of use if you have huge non-HCM configuration data, you can write a program (for example with REXX or C) to extract data from the tables you have exported with HCM and convert these data into a format that you can use for your applications. Or you can write a program which creates tables for the HCM format from your existing data for import.

Data format for importing and exporting

The import and export data are organized in tables. Each line of text in the table corresponds to a row of data in a database table or in a text file.

Each table type corresponds to an object that appears in an HCM diagram. Every line of text is processed. The tables contain a header line (the column definition record) with column names that describe the subsequent rows, the data records. A data record always begins with the table name followed by fields with data values. See also "Record structure."

HCM uses the first column (table name) and the second column (column name which in most cases is the object ID) to identify an object in the configuration. If the physical object does not exist, it is created. Logical objects cannot be created. Subsequent columns are used to update data values in the object. If an existing object is not mentioned in the import file, it remains unchanged in the configuration. This allows the import file to contain a subset of the objects in the configuration. Therefore, it is not possible to delete an object via the Import facility.

Some IODF objects are included in the import data format so that they can be properly placed within physical objects and that fields like descriptions and serial numbers can be updated. Furthermore, these records include a number of fields that will be ignored during the import operation. These fields are exported because they are useful for reports, but cannot be used as input except through HCM's user interface because they require extensive validation.

Example for a processor table:
```
"PROC","ID","Short Name","Model","Description","Serial Number"  1
"PROC","PROC1","P1","9021-822","Main Processor","1234569021"     2
"PROC","PROC2","P2","3090-600S","Old Processor","6543213090"     3
```

1   header line (column definition record)

2 and 3
   data record

File structure

The HCM Import and Export facility works with multiple files, one for each table. All files must be in the same PWS directory. The filenames are fixed 8 character table names. For a summary of the table names and for detailed table information, see "Table summary" on page 333 and "Detailed table formats" on page 335.

With the **Import Data** function, you can read multiple import files from a directory. For import-only, you can also combine one or more tables into a single file. Blank lines are not tolerated in files being imported. They result in the same error message as if column one of the line contained an unrecognized table name.

During export with the **Export Data** function, you are asked for the directory where the data tables should be created. Export writes each table in one file for the specified directory.

Record structure

Every row in a table represents one record. The first row of each table file is the column definition record (header line) which contains the column names. The column definition record defines the structure of the subsequent rows. The subsequent rows are the data records.

Each record contains several columns. Columns are separated with a character of your choice, for example, a comma or a tab. In this document a comma is used as a separator.

Compound names, such as partition names which are constructed from the processor name and the partition name are separated by a character of your choice, for example, a dot or a slash. In this document, a slash "/" is used as a separator. For example, PROC1/PART1 could also be PROC1.PART1.

The first column of each data record contains the table name, the second column contains the column name (in most cases it is the object ID) to identify a particular row. Subsequent columns contain the data for the description values of the table row.

For the import operation, it is required that each table starts with a column definition record. For the export operation, the column definition record is created by default, but you can suppress this behavior.

On import, each table can contain a subset of the columns in the table. The following examples could be used to update the description field of the processor (for example, PROC.TXT) and partition (for example, PART.TXT) tables, leaving all other values unchanged.

Example to update a processor table:
```
"PROC" 1 , "ID" 2 , "Description" 3
"PROC", "PROC1", "Main processor" 4
```

Example to update a partition table:
```
"PART" 1 , "ID" 2 , "Description" 3
"PART", "PROC1/LPAR1", "Test Partition" 4
"PART", "PROC1/LPAR2", "Production" 4
```

1 table name
2 column name (object ID)
3 column name (for example, description)
4 record data

The first reference to the table type must be the column definitions record. The column names are not processed in a case-sensitive way by HCM. If you read exported files using programs other than HCM, ensure that the programs are not case-sensitive when processing the table column names (column definition record).

The records for a particular table need not be contiguous in the input stream. For example, all column definition records (header lines) can be at the start of a file containing several tables. See the following examples of some possible ways your data can look like in a file containing several tables.

All column definition records appear at the beginning of the file:
```
"PROC", "ID", "Description"
"PART", "ID", "Description"
"PROC", "PROC1", "Main processor"
"PART", "PROC1/LPAR1", "Test Partition"
"PART", "PROC1/LPAR2", "Production"
```

The records can also appear in random order, as long as the column definition record precedes the record data:
```
"PART", "ID", "Description"
"PART", "PROC1/LPAR1", "Test Partition"
"PROC", "ID", "Description"
"PROC", "PROC1", "Main processor"
"PART", "PROC1/LPAR2", "Production"
```

or, more likely:

```
"PROC", "ID", "Description"
"PROC", "PROC1", "Main processor"
"PART", "ID", "Description"
"PART", "PROC1/LPAR1", "Test Partition"
"PART", "PROC1/LPAR2", "Production"
```

There must be only one column definition record for each table. Once the column definition records are known, subsequent column definition records are interpreted as data just like any other record.

An error message is issued by HCM if it encounters two or more records referring to the same object, but processing continues. For a given object, only the first record referring to it will be processed, all subsequent ones will be ignored.

Column formats

HCM recognizes six types of data columns:

Text   Text fields are free-format fields of a predefined length and can contain any sequence of keyboard characters. The maximum length of the text varies with each column. The text can contain imbedded spaces as well as characters that are used as separators. If you use, for example, a comma as a separator, you need to enclose it in double quotes (the usual quote used in prose, for example, ″A, and B″), or single quotes (the apostrophe character, for example, ′C, and D′). Use ″″ within a doubly quoted string to embed the double-quote character - ″He said ″″Hello″″.″. Single quotes can be used similarly, when single quotes are used to enclose the entire string. The Export function encloses text fields in double quotes.

Text fields are case sensitive. Fields which update IODF data must be entered in the appropriate case for the IODF field.

The maximum length of a text field is 254 characters.

> **Note:** As HCM Version 1.1.0 allows 255 character user fields, the last character may be truncated on export. A warning message is generated in such cases.

When updating existing records, a text field updates the configuration even if the field is all blanks or empty. Blank or empty is a valid value for a text field. If the entire column is not present in the input file, the field will be left unchanged.

Boolean (Bool)

These are fields with a Yes or No value. Accepted values in these fields are Yes, No, Y, N, 1, and 0 in upper or lower case.

A value enclosed in single or double quotes is also accepted although there is no need to do this. When updating existing records, a blank or empty Boolean field has no effect.

Numeric (Num)

Sequence of digits. Where appropriate, definitions of these columns will indicate acceptable ranges for these values. A value enclosed in single or double quotes is also accepted although there is no need to do this.

When updating existing records, a blank or empty Numeric field has no effect.

Hex   Sequence of hexadecimal digits (0..9, A..F). The field is checked that it is a

valid hexadecimal number. Where appropriate, definitions of these columns indicate acceptable ranges for these values.

A value enclosed in single or double quotes is also accepted although there is no need to do this. The processing of hex digits greater than 9 is not case sensitive, so an import accepts a..f, though an export produces A..F.

When updating existing records, a blank or empty Hex field has no effect.

Float Any floating point number which follows this generic format can be imported:

```
[whitespace] [sign] [ddd] [.] [ddd] [e|E[sign]ddd]
```

Where appropriate, definitions of these columns indicate acceptable ranges for these values.

A value enclosed in single or double quotes is also accepted although there is no need to do this.

When updating existing records, a blank or empty Float field has no effect.

On export, floats are expressed in this compatible format which avoids the use of exponential notation, with leading and trailing zeroes removed:

```
[ddd] [.] [ddd]
```

Enumerated (Enum)

These are fields that accept a single value from a predefined set of text values. As an example, physical ports on an ESCON director may be either "LED" or "Laser". Values are not case sensitive, i.e. Laser and LASER are equivalent.

A value enclosed in single or double quotes is also accepted although there is no need to do this.

When updating existing records, a blank or empty Enumerated field has no effect.

In all cases, leading and trailing spaces are ignored.

Hierarchical object keys

Where applicable, the object keys in the tables are structured in a hierarchical manner. This has the following benefits:

- The key itself, with the object type implied by the table containing the data, is sufficient to completely identify the object without reference to other tables to obtain the contained-in relationships. This simplifies reporting.
- A sorted list of table rows groups related objects together. For example, the Controller Channel Interface (CCI) table would contain the following object keys:

```
3990 #12/SC0/A
3990 #12/SC0/B
3990 #12/SC1/A
3990 #12/SC1/B
```

- A database query could collect items using keys containing wild cards. For example, from the Controller Channel Interface (CCI) table:

```
3990 #12/*          All channel interfaces on 3990 #12
3990 #12/SC0/*      All channel interfaces on segment SC0 of
                    3990 #12
```

Unique key

All objects which can be mentioned as a source or target in the CONNECT or LOGLINK tables have a unique key field. This field is for export-only and contains a combination of the table name and the object ID. This is necessary since the object ID by itself is often not sufficient to uniquely identify the object. For example, A1/F2 could be CHPID F2 of processor A1, or it could be port F2 of switch A1.

In the case of patchports and daisy-chained parallel interfaces, the source or target fields in the CONNECT table are qualified with a suffix indicating exactly which of the multiple interfaces is being referred to. (See discussion under the Table 35 on page 359). In these cases, a simple concatenation of the table name and the object ID is not sufficient to use as a key. Using the explicit unique key for database joins, and the source and target fields as the descriptive text, works well and is easy to describe to the database platform when using the tables exported by HCM.

Future compatibility and extensibility

The file format is flexible to easily accommodate additional tables and/or fields in the future. Older files without the new tables and fields would still be importable by later versions of HCM.

Working with the import and export function

To import and export your configuration data from text files to your configuration, open **Import Data...** or **Export Data...** from the **File** menu.

Use a text editor that processes more than 80 characters per line. Or you can use a spreadsheet program, for example, Lotus 1-2-3.

For more information on the data formats, table names, and table details, refer to "Data format for importing and exporting" on page 317, "Table summary" on page 333, and "Detailed table formats" on page 335.

Importing data

With the import function you can populate data of non-HCM applications or processes into HCM to do the following:

- Setup a new HCM configuration file with existing physical data.

  To do this, select **Create a new file** in the **Choose HCM file for Import** dialog. If you import data into an empty file, you are doing an initial setup of the HCM configuration file. For example, you are importing infrastructure data. In this case, the IODF must exist and the IODF definitions should match the infrastructure data.

- Update an existing HCM configuration with your data.

  To do this, select **Open an existing file** in the **Choose HCM file for Import** dialog. If you import data into an existing configuration file, you are updating it with your infrastructure data. You can, for example,
  – create cabinets, patchports, crossbar switches,
  – provide new or changed serial numbers or descriptions,
  – import cables and place them into connections,
  – provide VOLSERs of devices,
  – create and delete port-CHPID, port-port, and port-CU links.

You can either import all the data in a table, or you can import a subset of the table. For example, when it is not necessary for all columns or objects to be imported, you can import just the ones you need.

The fields and columns mentioned in the import stream must match current values, or an update is done. Blanks and 0 are valid values. Some table fields cannot be imported. They are ignored.

You cannot delete an object using the Import function.

Tip

To create the tables for import,
- you can convert your data to the HCM format with a program (for example, C or REXX), or
- use the export function first, then edit the tables, and import the changed tables into HCM.

The data format for importing, for example, how each row of data in the file corresponds to an object in the configuration is explained in the chapter "Data format for importing and exporting" on page 317.

Choose **Import Data...** from the **File** menu. (This menu option is disabled, when an HCM configuration file is open.) A confirmation message is displayed which informs you that the import operation is a major update to your configuration and that it cannot be undone. Therefore, this message recommends that you save your configuration before importing data.
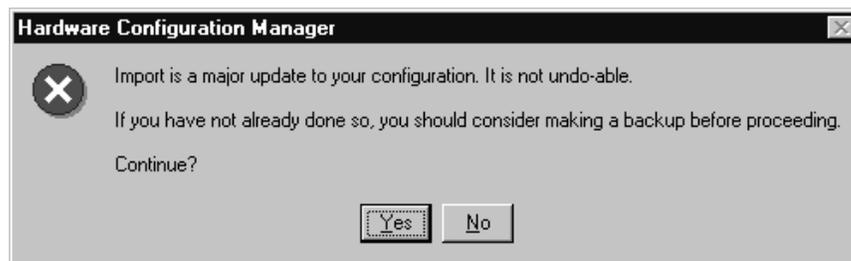
Figure 150. Hardware Configuration Manager Confirmation Message

No terminates the import and returns to the main menu. Choose **Yes** (the default) to continue. The **Choose HCM file for Import** dialog appears.

Figure 151. Choose HCM File for Import Dialog

You can choose to update an existing HCM file or to create a new one. When importing against an existing HCM file, the IODF may or may not be in sync with the HCM file. Therefore, it is possible to build, or add to, an HCM file by processing input from *both* the import file and an IODF concurrently.

Note: The option **Create a new file** is disabled when you are running in stand-alone mode.

Select **Open an existing file** (this is the default) and click on **OK**.

(You can also select **Create a new file**. Then, the **New** dialog appears instead. Specify a filename and click **OK**. The **High-level Qualifier** dialog appears where you can confirm or specify a high-level qualifier before the list of IODFs appears in the **Select IODF** dialog.)

The dialog **File Open** appears. Select an existing HCM file for update and click on **Open**.

Figure 152. File Open Dialog

On the following **Choose IODF** dialog, you can select if you want to work with the already associated IODF or if you want to work with an IODF that is not associated with the opened HCM file.

General Concept on IODF Update Processing: The fields which update the IODF receive special handling, that is, IODF objects which do not exist in the configuration yet, cannot be created and the following update types must be confirmed:

• Serial numbers in Processor, Switch, Control Unit, and I/O Device objects.
• Descriptions in Processor, Partition, CHPID, Switch, Control Unit, and I/O Device objects.
• VOLSERs in I/O Device objects.
• Labels in Port objects.
• Creation and deletion of Port-CHPID, Port-Port, and Port-CU links.

There will be one dialog for each of these fields. The dialogs only appear as needed.

Note: There could be several thousand items in each list. In such cases, the list box will not be able to contain all of the items. Should this occur, the user will be informed that the dialog will appear repeatedly to process the updates in "batches". Each batch will contain several thousand items.
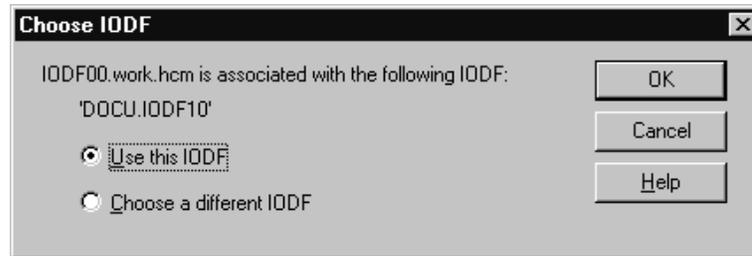
Figure 153. Choose IODF Dialog

Select the option **Use this IODF** and click on **OK**. Now the HCM file and the IODF are known. The **Choose Import Files** dialog appears.

(You can also select **Choose a different IODF**, when you want to use an IODF that is not yet associated to the HCM file. Then, the **High-level Qualifier** dialog appears where you can confirm or specify a high-level qualifier before the list of IODFs is displayed in the **Select an Alternate IODF** dialog.)
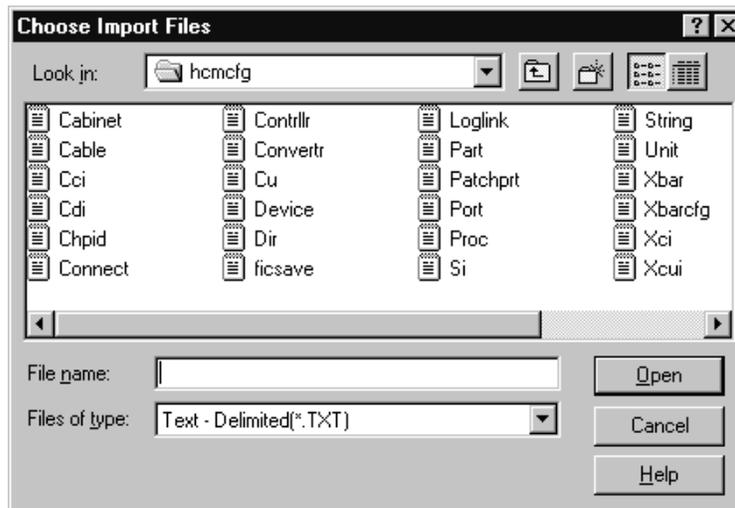
Figure 154. Choose Import Files Dialog

You can select one, multiple or all import files that appear in the list box in the specified directory that you want to use for your configuration.

The **Files of type** list contains a (default) entry suitable for comma-delimited import files with the file extension *.csv, as well as text-delimited (*.txt) files and the All files *.* entry.

Click on **Open** if you have selected at least one file for import. The **Text File Options** dialog appears.
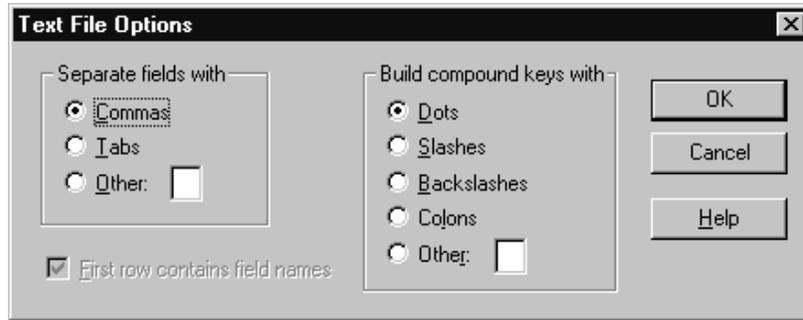
Figure 155. Text File Options Dialog

On this dialog you can specify how the import file should be interpreted by HCM. In the section **Separate fields with** you can specify the separator to be used between fields in each record. The **Other** field is one character long. When you type data into this field, the **Other** radio button is automatically selected. When you select one of the other radio buttons it clears the **Other** field.

In the section **Build compound keys with** you can specify the separator between parts of compound names such as a CHPID name (Processor name {separator} CHPID number).

Examples:
```
PROC1.30
PROC1/30
```

In the **Other** field, HCM will not allow a parenthesis symbol ″(″ or ″)″ as a separator for compound names.

The **First row contains field names** check box is disabled. This checkbox is always selected and left disabled when importing data, because all import files must have column definition records.

The actual import processing occurs immediately after you have clicked on **OK** in the **Text File Options** dialog. The status bar shows progress information.

Cancel terminates the import operation and returns to the main menu.

However, HCM examines the import sources to determine if the IODF needs to be updated. If updates are required, for example, the **Proposed Description Updates** dialog appears where you can confirm the updates.
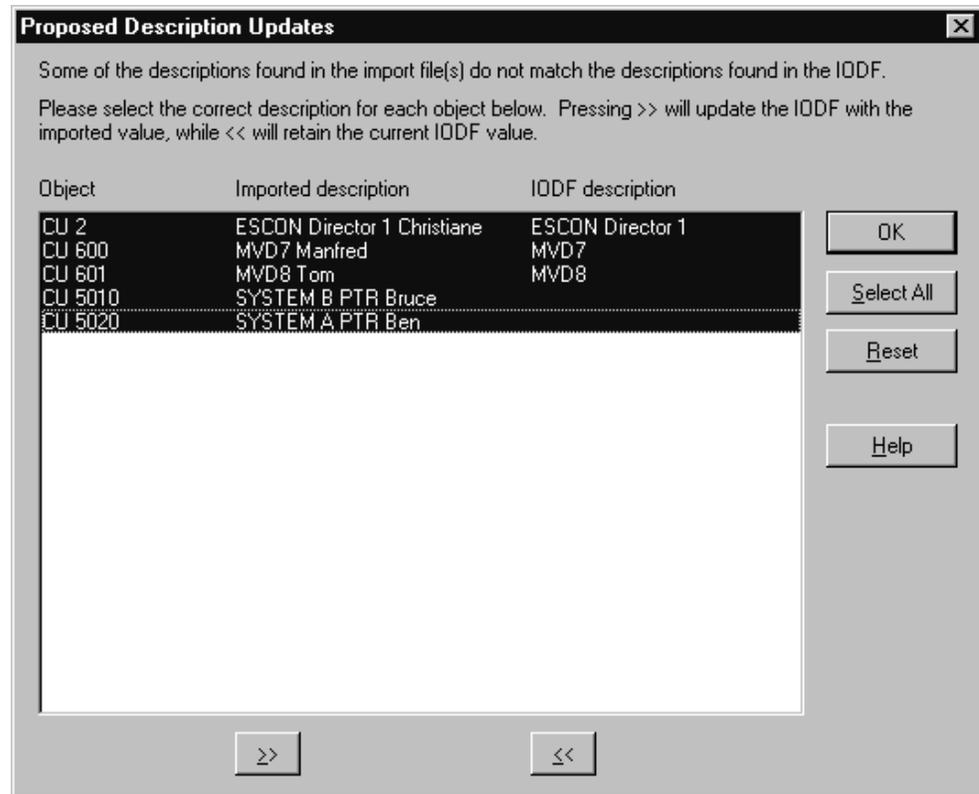
Figure 156. Proposed Description Updates Dialog

The list box shows one line for every object in the configuration for which the description value in the IODF and in the import files differ. On entry, all the lines are preselected. You can select one or more objects.

You can process lines in the list box through the use of the push buttons >> and <<. Once a line has been processed, HCM grays it. Processing a line means that the direction of the intended transfer is set as follows:

- pressing **>>** will update the IODF with the imported value
- pressing **<<** will retain the current IODF value.

Each line of the list box shows the object identifier, the value of the description field in the import files read by HCM, and the value of the description field in the IODF on the host.

The dialog remains displayed even if all lines have been grayed, because the **Reset** button allows you to change the value of assignment for already processed lines. This button can be used to mark selected lines as requiring processing again, which removes their gray color and the >> or << symbol from those lines. This button unprocesses the selected processed lines.

Click on the **OK** button when you are satisfied with the transfer for each processed line. HCM will then proceed to update the local and/or the remote database. If, however, any unprocessed lines remain in the list box, clicking **OK** results in an error message stating that you must process all lines in the list box before clicking **OK**.

Port <> CHPID/PORT/CU Links:

The normal behavior of HCM in the dialog when making or breaking a connection involving a switch port is to update the IODF with the appropriate IODF links. Making or breaking a connection via the import file will behave in a similar fashion, with the exception that you will have the opportunity to overwrite the IODF update.

Note: Physical connections are always made in the configuration file even if the IODF definitions are not updated.

A dialog will appear to display and confirm the Port link updates. For each port affected by the import, it will indicate the current state in the IODF and the proposed changes from the import file. For each affected port, all port links will be displayed so that you can see the information in context.

Updating the Controller Serial Numbers:

The serial numbers assigned to control units come from the CONTRLLR records in the import file, not the control unit records. Control unit records have a serial number field, but this data is used for export-only. When a serial number is imported for a controller, HCM attempts to ensure that all control units in the controller have that serial number already. If there are any that do not, HCM suggests updates for those control units. If you accept the IODF values for the control unit serial numbers (rather than the imported values), the control units' serial numbers remain unchanged but the controller itself receives the new serial number.

Note: This can also occur as a result of a rejected host update which forces you to accept existing IODF values.

At the end of the import process, HCM steps through all controllers and ensures that there are no mismatches between the controller serial number and the serial numbers of control units contained in the controller. This is done in two steps:

1. If all the CUs in a controller have the same serial number, but the controller's serial number is different, the controller's serial number is changed to match the CUs.

2. If the CUs within a controller have serial numbers that differ from each other, CUs are moved out, one at a time, into other controllers. To avoid serial number conflicts, new controllers are created if necessary to hold the CUs moved out.

After moving out a CU, step 1 above is applied again, then step 2, repeating until there are no remaining conflicts. Thus, if you reject the suggested imported control unit serial numbers for any reason, it may result in the creation of additional controllers.

If your import is complete, an informational message is shown.

Exporting data

Choose **Export Data...** from the **File** menu. This menu option is only enabled, when a HCM configuration file is open. The **Export Data** dialog is opened.

Tip

Export to an empty directory. A non-empty directory can already contain exported files from a previous export operation. If you do not choose to export exactly the same tables in the current export operation, some of the old files will continue to reside in the directory without being overwritten. Mixing files in the same directory from differing export operations can mislead you into attempting to use exported files which do not represent the same state of the configuration at all.
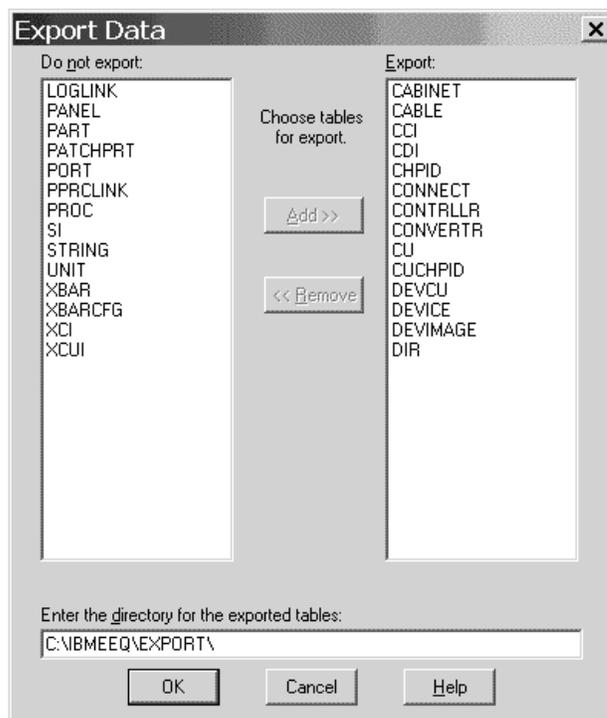
Figure 157. Export Data Dialog

Specify the tables to be exported, as well as the directory where the exported tables will be created. The specified directory must already exist.

The dialog is initialized based on values saved in the EEQHCM.INI file from the last execution of this dialog. The first time that you bring up this dialog, all tables will be enabled for export except for DEVCU, CUCHPID, and DEVIMAGE. The reasons for not exporting the three above mentioned tables by default are:
1. These tables deal with IODF data only.
2. They may be extremely large (especially the DEVCU and DEVIMAGE tables).

Select one or more tables in the **Do not export** section, click on **Add** (or double-click an entry) to move any selected tables from the left list box into the right list box to be included for export.

If you want to remove one or more tables from the **Export** section, select one or more tables, click on **Remove** (or double-click an entry) to move the tables to the **Do not export** section.

Click **OK** to accept the list box contents.

If any tables will be overwritten, you are asked if you want to replace the existing files. You can confirm with **OK** or reject with **Cancel**.

If no files will be overwritten, or you confirmed the query about replacing files with **OK**, the **Text File Options** dialog appears.
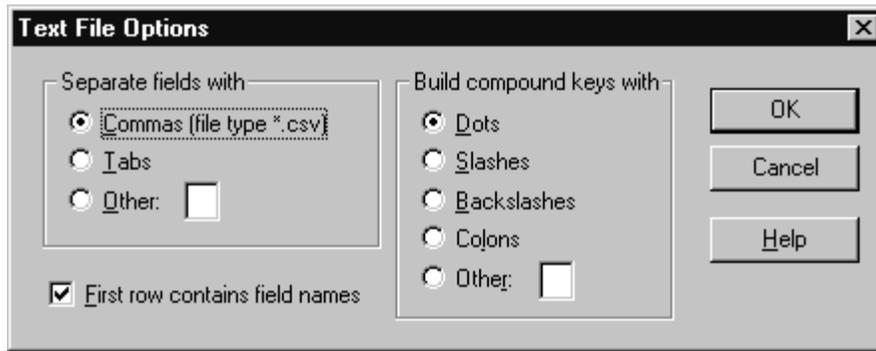
Figure 158. Text File Options Dialog

In this dialog you can specify how the export file should be constructed regarding the separators. The dialog is initialized with the values last used for import or export in the current HCM session. The defaults on first use in each session are commas, dots, and the **First row contains field names** is checked. In the section **Separate fields with** you can specify the separator between fields in each record. When you type data into the**Other** field, the adjacent radio button is automatically selected.

In the section **Build compound keys with** you can specify the separator between parts of compound names such as a CHPID name (Processor name {separator} CHPID number). For example, PROC1.30, PROC1/30. Because of the way tags are constructed for unlabeled strings, HCM will not allow a left or a right parenthesis as a separator for compound names.

The **First row contains field names** check box is enabled when you are exporting. If it is checked, HCM ensures that the first line in all export files is the column definition record.

Click on **OK** to perform the export. The status bar will contain progress information.

HCM informs you with an informational message that exporting completed successfully.

Importing in stand-alone mode

Import is allowed in stand-alone mode, or with read-only access to the IODF. The tables can be imported to work configuration files.

An import operation that requires an update to the IODF is not possible. If an input record is encountered which would require an update to the IODF, for example, updating the description of a CHPID, an error message is issued. In other words, you can include CHPID records with descriptions, as long as those descriptions match the current description of the CHPID. The entire record is not

discarded, just the data columns which required update access to the IODF. In all cases, update access to the HCM file is required.

In particular, importing of VOLSERs is permitted in stand-alone mode, though this may cause appearance of the VOLSER Reconciliation dialog the next time the configuration is opened on-line. For more information see "VOLSER reconciliation" on page 94.

Exporting in stand-alone mode

Exporting is possible in stand-alone mode.

Some tables cannot be exported in stand-alone mode from a work configuration file, because they contain data which is only available from the IODF (DEVCU, CUCHPID, DEVIMAGE).

All tables can be exported from a production configuration file.

Using import as a migration aid

Many of the fields, and tables are optional when a new configuration file is being created from an import file and an IODF. This section provides guidelines of what is really necessary.

PROC Entries in this table need only be provided when updating or specifying attributes for these objects. Alternate processors, and processors which are in the "One Processor Per Partition" style need to appear in the import file. In the normal case, processors are created based on the IODF.

PART Entries in this table need only be provided when updating or specifying attributes for these objects. Partitions are created based on the IODF.

CHPID
Entries in this table need only be provided when updating or specifying attributes for these objects. CHPIDs are created based on the IODF.

DIR (Switches)
Entries in this table need only be provided when updating or specifying attributes for these objects. Switches are created based on the IODF.

PORT (Switch Ports)
Ports which have a LASER interface instead of a LED need to appear in the import file. All others are created (as LED interface) based on the IODF.

CONTRLLR
Controllers which need to be associated with a specific physical description (for example, 3990s) need to appear in the import file. Controllers which are grouped with other controllers (for example, Dual Frame, and tape controllers grouped with communicator cables) need to be mentioned in the import file. All controllers mentioned in the CONNECT table must appear in the import file.

CCI (Controller Channel Interfaces)
For controllers which have more than the minimum number of channel interfaces allowed by their physical description file, all interfaces beyond the minimum need to appear in the import file. These records are also necessary to specify the interface type if the default interface type for the controller is not appropriate.

CDI (Controller Device Interfaces)

These can usually be omitted. For controllers which have more than the minimum number of device interfaces allowed by their physical description file, all interfaces beyond the minimum need to appear in the import file. Controllers with a variable number of device interfaces are very rare.

CU (Control Units)

CUs which are contained in a controller which has more than one segment should appear in the import file. This ensures that the load/resync process will place the CU into the proper segment. If a proper serial number has been assigned to a control unit in the IODF, it need not appear in the import file.

STRING

Strings which need to be associated with a specific physical description (for example, 3880s, RAMAC®) need to appear in the import file. All strings mentioned in the CONNECT table must appear in the import file.

SI (String Interface)

This table can usually be omitted. The minimum number of string interfaces are created as necessary based on the physical description.

UNIT   Entries in this table need only be provided when updating or specifying attributes for these objects. Units are created as necessary based on the physical description.

DEVICE

Where the string which contains this device has been included in the import file, the first DEVICE in the string should also appear in the import file.

XBAR   All crossbar switches you wish to create need to appear in the import file.

XCI (Crossbar Channel Interface)

The import operation does not read this table.

XCUI (Crossbar Control Unit Interface)

The import operation does not read this table.

XBARCFG

Entries in this table need only be provided when updating or specifying attributes for these objects. This table can be omitted.

CABINET

All cabinets you wish to create need to appear in the import file.

PANEL

All panels you wish to create within a specific cabinet.

PATCHPRT

Entries in this table need only be provided when updating or specifying attributes for these objects. Patchports are created based on the information in the CABINET table.

CONVERTR

All converters mentioned in the CONNECT table must appear in the import file.

CABLE

All named cables mentioned in the CONNECT table must appear in the Import file.

LOGLINK
> Entries in this table need only be provided when updating or specifying attributes for these objects.

CONNECT
> Each connection you wish to create must appear in the import file. (The load/resync process may create additional connections as well.)

PPRCLINK
> All PPRC links/PPRC connections you wish to create, need to appear in the import file.

Table summary

Many of these tables are optional. See "Using import as a migration aid" on page 331 for a discussion of the tables required to create a configuration using import. The DEVCU, CUCHPID, and DEVIMAGE tables are provided to facilitate reporting of logical connectivity based on exported files.

The table names have been limited to 8 characters to maintain a one to one correspondence between the table names and the filenames used by the export process, and by defining the database tool reading the exported data. For example, the PATCHPRT table is written to PATCHPRT.TXT.

Table 11. Table Summary

Category	Table name	Description
Processor related	**PROC**	a processor, see "PROC table" on page 335
	PART	a partition, see "PART table" on page 337
	CHPID	a CHPID, see "CHPID table" on page 337
Switch related	**DIR**	a switch, see "DIR table" on page 339
	PORT	a switch port, see "PORT table" on page 340
Controller/Control unit related	**CONTRLLR**	a physical controller, see "CONTRLLR table" on page 340
	CCI	a controller channel interface, the interfaces on the top of a controller: A, B, C ..., see "Controller channel interface (CCI) table" on page 343
	CDI	a controller device interface, the interfaces on the bottom of the controller: SP0, SP1, ..., see "Controller device interface (CDI) table" on page 344
	CU	an IODF control unit, see "CU table" on page 345
String/Device related	**STRING**	a physical string, see "STRING table" on page 346
	SI	a string interface, the interfaces on the top of a string: A0, A1 ..., see "String interface (SI) table" on page 348

Table 11. Table Summary (continued)

Category	Table name	Description
	UNIT	a physical unit of a string, see "UNIT table" on page 348
	DEVICE	an IODF I/O device, see "DEVICE table" on page 349
Crossbar switch related	**XBAR**	a physical crossbar switch, see "XBAR table" on page 350
	XBARCFG	a named crossbar configuration describing the crossbar connectivity, see "XBARCFG table" on page 351
	XCI	a crossbar channel interface (not read by the import function), see "Crossbar channel interface (XCI) table" on page 351
	XCUI	a crossbar control unit interface (not read by the import function), see "Crossbar control unit interface (XCUI) table" on page 351
Patchport related	**CABINET**	a patchport cabinet, see "CABINET table" on page 351
	PANEL	a patchport panel, see "PANEL table" on page 352
	PATCHPRT	a patchport, see "PATCHPRT table" on page 353
Connection related	**CABLE**	a physical cable, see "CABLE table" on page 355
	CONVERTR	a converter, see "CONVERTR table" on page 354
	LOGLINK	describes the end points of a logically complete series (see "LOGLINK table" on page 356). Useful for reporting purposes. See "LOGLINK table" on page 356.
	CONNECT	a connection between two objects, see "CONNECT table" on page 359
	PPRCLINK	a PPRC link/PPRC connection, see "PPRCLINK table" on page 362
Logical connection related	**DEVCU**	device - control unit connections, see "DEVCU table (export-only)" on page 362
	CUCHPID	control unit - CHPID connections, see "CUCHPID table (export-only)" on page 362
	DEVIMAGE	device - processor image connections (an image being a logical partition or the single image in a BASIC mode processor), see "DEVIMAGE table (export-only)" on page 363

Detailed table formats

Fields which are always required are in **boldface type**. All other fields are optional, at least when updating an object. Optional fields which are required when creating an object are noted as such in the column description.

The tag **EO** means export-only and indicates that the field is filled in by export, and is read in by import but is not used to update the object. These fields contain IODF data.

Data columns which update the IODF are marked with **<Updates the IODF>**.

Objects which must be present in the IODF are marked with **<IODF Key>**.

Fields which are ignored in certain contexts, are still validated for correctness (for example, any data supplied in a numeric column must be numeric).

For a number of fields, a default value is indicated. When importing, it is important to realize that the default value is only used when a new object is being created and when no data is provided in the import file for that field.

PROC table

Table 12. PROC Table

Field Name	Format	Description
Table Name	**Text**	**PROC**
ID	Text	<IODF Key> The ID of the IODF object. (This is the processor's name.)
Grouped with	Text	Applicable only for new processors that are not already in the PWS configuration.
		For alternate processors, and OPPP (One Processor Per Partition) style processors, the name of the primary processor that this processor should be grouped with.
		Ignored for processors which are already in the PWS configuration file (they have already been placed).
Style	Enum	Applicable only for new processors that are not already in the PWS configuration.
		If *Grouped with* is specified:
		• This field defaults to the style of the *Grouped with* processor.
		If specified here, it must match the style of the *Grouped with* processor.
		If *Grouped with* is blank, or contains your processor name (grouped with self):
		• This field defaults to IBM.
		• Either value may be coded.
		Valid values are IBM or OPPP (One Processor Per Partition). Ignored for processors which are already in the PWS configuration file.

Table 12. PROC Table (continued)

Field Name	Format	Description
Status	Enum	Applicable only for new processors that are not already in the PWS configuration. If *Grouped with* is specified and that processor is IBM style: • Alternate is the only valid value. • This field defaults to alternate in this case. If *Grouped with* is specified and that processor is OPPP style: • Either value is valid. • This field defaults to alternate in this case. If *Grouped with* is blank, or contains your processor name (grouped with self): • Primary is the only valid value. • This field defaults to primary in this case. Valid values are: primary, alternate. Ignored for processors which are already in the PWS configuration file.
Short name	Text	The short name of the physical machine which contains this processor. Defaults to the first two characters of the processor ID (or first character in the case of single character IDs). Export will place the same value in this column for all processors in the physical machine. On import, all processors in the machine are expected to have the same value for this attribute, but this is not strictly enforced. If the specified value does not match that of other PROC records in the import file having the same machine, the last PROC record found in the import file will be taken. You will *not* be informed with a warning message. Valid values: any string of one, or two text file characters in which each character must be either non-alphabetic or uppercase alphabetic. An empty string is not a valid value.
Serial number	Text	<Updates the IODF> The serial number of the IODF object.
Description	Text	<Updates the IODF> The description of the IODF object.
Model	Text EO	Value taken from the IODF object. For example, 9021-720.
Support level	Text EO	Value taken from the IODF object.
Mode	Text EO	Value taken from the IODF object. For example, BASIC.
Network name	Text EO	Value taken from the IODF object.
CPC name	Text EO	Value taken from the IODF object.
User field 1	Text	User field 1
User field 2	Text	User field 2

Table 12. PROC Table (continued)

Field Name	Format	Description
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

PART table

Table 13. PART Table

Field Name	Format	Description
Table Name	**Text**	**PART**
ID	Text	<IODF Key> The ID of the IODF object. This consists of the processor name followed by a slash ″/″ and the partition name.
Description	Text	<Updates the IODF> The description of the IODF object.
Number	Text EO	Value taken from the IODF object. The partition number in hex. Blank if not applicable.
Usage	Text EO	Value taken from the IODF object. For example, CF.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

CHPID table

Table 14. CHPID Table

Field Name	Format	Description
Table Name	**Text**	**CHPID**
ID	Text	<IODF Key> The ID of the IODF object. This consists of the processor name followed by a slash ″/″ and the hex CHPID number.
Unique Key	Text EO	The unique key for this object. Composed of ″Table Name:ID″
Description	Text	<Updates the IODF> The description of the IODF object.
Type	Text EO	Value taken from the IODF object.
Mode	Text EO	Value taken from the IODF object. For example, SHR.
Dynamic switch	Text EO	Value taken from the IODF object. Blank if not applicable. For example, 0F.

Table 14. CHPID Table (continued)

Field Name	Format	Description
Physical number	Hex	The physical channel onto which this CHPID is mapped. Defaults to the CHPID number.
		This is only applicable for OPPP style processors, and need only be specified if the logical CHPID is mapped onto a different physical channel.
		In cases where the CHPID has no corresponding physical object, 0 will appear here on export. It is not possible to import CHPIDs without a corresponding physical object. Valid values are: 0, 1, 2, ..., 1FF
Interface Type	Enum	One of HCM's allowed CHPID interface types.
		Defaults based on the CHPID type of the CHPID in the IODF.
		This need only be specified if the physical interface type is not compatible with the logical CHPID type.
		Valid values are: serial, parallel, coupling, IOC, OSA.
PCHID	Text EO	Value taken from the IODF object. Applicable only for XMP processors. Blank if not applicable.
Spanned with	Text EO	If the CHPID is defined as spanned, this column contains all channel subsystems the CHPID is spanned over. Applicable only for XMP processors. Blank if not applicable.

Sparse CHPID mapping for OPPP processors

When processors are defined using the One Processor Per Partition (OPPP) style, several logical processors can share the same physical hardware, including the limited set of channel paths physically present on the machine. A one-to-one association of logical CHPIDs to physical channel paths is set up on a per processor basis. Usually, CHPID *nn* maps to path *nn*, regardless of which processor CHPID *nn* belongs to. However, it is possible to make non-default associations such that, for instance, CHPIDs 00 through 1F for PROC1 map to paths 00 to 1F, while CHPIDs 00 through 1F for PROC2 map to paths 100 to 11F.

Any non-default CHPID-to-path mapping can be represented in the CHPID record by filling in the "Physical number" field. It is only possible to represent CHPID-to-path mapping for CHPIDs which actually exist in the configuration, because CHPID import records are rejected by HCM unless the CHPID exists. When editing an existing configuration using the HCM interface you can specify mappings for CHPIDs which have yet to be created. This cannot be done using the Import function.

Example: If you were to map CHPID 00 to path 10, CHPID 00 must exist in the configuration. If CHPID 10 existed, you could also specify a path for it to map to. Assuming, however, that CHPID 10 does not exist, no mapping can be specified for it in an import file, and HCM will automatically assign a mapping to it that does not conflict with any other mappings. If the next free path number (after 10) that is free is 100, then CHPID 10 will be mapped to path 100. (100 is a likely choice, because HCM looks for an unmapped path starting with the path that CHPID 10 used to be mapped to, in this case 10. All the paths up to FF are spoken for due to the default mapping.)

```
CHPID 00  ====>  Path 10   assigned by CHPID record in import file
CHPID 10  ====>  Path 100  assigned by default (first free path)
```

This is known as sparse CHPID mapping, because the import file does not contain a record to map every CHPID slot to a path slot. Some CHPID slots are allowed to default. Export practices sparse CHPID mapping on output.

OPPP and shadow processor CHPID/path mapping issues

If a configuration contains shadow processors or One Processor Per Partition style processors, it is possible for more than one CHPID to use the same physical channel path. LOGLINK and CONNECT records mention CHPIDs when identifying a physical channel path. A problem arises if a set of import/export files describes more than one connection to a path. Therefore, during export, LOGLINK and CONNECT records are only generated once for each path, not for each CHPID using the path.

If you are generating files for import using tools other than HCM's export use the same approach. That is, you should not generate duplicate LOGLINK and CONNECT records when the configuration contains paths which map to multiple CHPIDs.

DIR table

Table 15. DIR Table

Field Name	Format	Description
Table Name	Text	DIR
ID	Hex	<IODF Key> The ID of the IODF object.
Serial number	Text	<Updates the IODF> The serial number of the IODF object.
Description	Text	<Updates the IODF> The description of the IODF object.
Model	Text EO	Value taken from the IODF object. For example, 9032-3.
Controller Label	Text	Applicable only for new switches that are not already in the PWS configuration. The label which should be applied to the internal controller of this switch. Ignored for switches which are already in the PWS configuration file.
String Label	Text	Applicable only for new switches that are not already in the PWS configuration. The label which should be applied to the internal string of this switch. Ignored for switches which are already in the PWS configuration file.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

PORT table

Table 16. PORT Table

Field Name	Format	Description
Table Name	**Text**	**PORT**
ID	Text	<IODF Key> The hex switch number followed by a slash "/" and the hex port number.
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:ID"
Interface type	Enum	One of HCM's allowed Port interface types. Valid values are: LED, Laser.
Name	Text	<Updates the IODF> The descriptive name of the IODF port.

Note: Ports which are not "installed" in the IODF are not exported, and cannot be imported.

CONTRLLR table

Table 17. CONTRLLR Table

Field Name	Format	Description
Table Name	**Text**	**CONTRLLR**
Label	Text	Unique label of an HCM physical controller. For example, 3990 #12.
Serial number	Text	<Updates the IODF> In addition to updating the controller object (a PWS object), this field will be used to update the IODF for all control units that the controller contains.
IODF control unit type	Text	The type-model of the controller. For example, 3990-6. Applicable only for new controllers that are not already in the PWS configuration. Ignored for controllers which are already in the PWS configuration file. Export places the following in this field: The type-model of one of the control units in the controller. The assumption is that all control units in the controller are of the same type. Where the controller contains more than one control unit, the specific control unit chosen is undefined. If the controller contains no control units, this field will be blank. See the "Usage Note" on page 341 in the PDF Name column.

Table 17. CONTRLLR Table (continued)

Field Name	Format	Description
PDF AbbrevId	Text	The abbreviated ID (AbbrevId =) of one of HCM's physical description templates. For example, RAMAC2. Applicable only for new controllers that are not already in the PWS configuration. Ignored for controllers which are already in the PWS configuration file. Export places the following in this field: If a specific PDF was used, and that PDF has an abbreviated ID, it will appear here. Otherwise this field will be blank. See the "Usage Note" on page 341 in the PDF Name column.
PDF Name	Text	The label (Subsystem =) of one of HCM's physical description templates. For example, IBM 3990 Model 6 DASD Controller. Applicable only for new controllers that are not already in the PWS configuration. Ignored for controllers which are already in the PWS configuration file. Export places the following in this field: If a specific PDF (physical description file) was used, the physical description name (Subsystem =) will appear here. Otherwise this field will be blank. **Usage Note:** Import usage of the IODF Control Unit Type, PDF AbbrevId, and PDF Name fields: These fields are used to determine which HCM physical description should be associated with a new controller. Only one of these fields is used on input. They are evaluated in the following order: 1. PDF name is used if the column is present and contains data. If specified it must be valid. 2. PDF AbbrevId is used if the column is present and contains data. If specified it must be valid. 3. HCM's normal search for a PDF based on the IODF Control Unit Type field. If the IODF Device Type is blank or invalid, the default PDF is assigned.
Display class	Enum	One of HCM's controller display classes. If the controller is being created, and this field is not specified, it will default as follows: 1. if the PDF specifies a display class, that class will be used. 2. the display class will default based on the first control unit that is placed into this controller. 3. if neither of the above apply, the display class will default to "Other". Valid values are: DASD, Tape, Terminal, Unit record, Telecommunications, MICR/OCR, Graphics, CTC, Other.
Installed	Bool	Yes (this is the default) indicates that the controller should be marked as installed.

Table 17. CONTRLLR Table (continued)

Field Name	Format	Description

Integrated	Bool	Yes indicates that the controller should be marked as containing integrated devices (a 3745, for example). The default is based on the display class:
		    DASD and Tape: No
		    All others: Yes

Grouped with	Text	Applicable only for new controllers that are not already in the PWS configuration. For controllers which can be grouped together with communicator cables (3803 and 3480 for example), and Dual Frame controllers, this column specifies the Controller Label of the leftmost controller in the subsystem. This field is valid only for controller with PDFs which allow more than one controller in a subsystem. Ignored for controllers which are already in the PWS configuration file (they have already been placed).
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

Interfaces on controllers

When controllers are created with the import facility, each controller is created with the minimum number of channel and device interfaces. This is different from normal HCM behavior. Normally HCM would create the controller with the default number of interfaces as specified in the PDF. The minimum number of channel interfaces is the lowest value specified in the Allow clause of the PDF, or 1 for the default PDF. Additional interfaces are created by mentioning them in the CCI table.

Note that it is possible with HCM to decrease the number of interfaces below the minimum number of interfaces specified in the PDF Allow clause. For example, it is possible to decrease the number of interfaces on a 3990 segment so that each segment contains only one interface. It is not possible to create this configuration with import without altering the PDF to allow a minimum of 1 interface on a 3990. This is a known instance where an export, followed by an import into an empty file, will not recreate the original configuration exactly.

Controller channel interface (CCI) table

Table 18. CCI Table

Field Name	Format	Description
Table Name	Text	CCI

Interface name	Text	The name of the controller channel interface, in the following form:
		
		controller label[/segment name][/adapter name]/interface label
		controller label
		    label of controller
		segment name
		    segment label (used only with controllers having more than one segment)
		adapter name
		    channel adapter label (used only with controllers using channel adapter scheme)
		interface label
		    channel interface label (A, B, ...)
		
		In cases where the components (segment, channel adapter, or interface) are not given explicit names in the PDF, a zero-based index is used instead. (for example, channel adapter 2 is the third channel adapter in the segment or controller which contains it).
		
		Examples:
		`3480 #9/A`
		`3990 #12/SC0/A`
		`6100 #5/Bus 0/A/0`

Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Interface name".

Interface type	Enum	One of HCM's allowed interface types.
		
		The default is based on the display class of the controller:
		    DASD and tape: serial
		    All others: parallel
		
		Note: If the interface type of an existing interface is being changed, this interface will be disconnected if necessary before being updated. This is necessary since the interface type change will mean that the cable currently connected to this interface will no longer be of the correct type.
		
		Valid values are: serial, parallel, IOC, OSA.

Note: All controller channel interfaces beyond the minimum must be explicitly created with this record. If a CCI must be created, all intervening CCIs and channel adapters will be created as necessary. For example, creating the fourth CCI will automatically create the second and third if they do not already exist. In this example, the interface types of the second and third interfaces would be assigned so that they match the type of the fourth interface.

Controller device interface (CDI) table

Table 19. CDI Table

Field Name	Format	Description
Table Name	**Text**	**CDI**

Interface name	Text	The name of the controller device interface, in the following form:
		
		controller label[/segment name][/adapter name] /interface label
		controller label
		    label of controller
		segment name
		    segment label (used only with controllers having more than one segment)
		adapter name
		    device adapter label (used only with controllers using device adapter scheme)
		interface label
		    device interface label. When there is only one device interface on the adapter or segment, the interface label may be omitted.
		
		In cases where the components (segment, device adapter, or interface) are not given explicit names in the PDF, a zero-based index is used instead. For example, device adapter 2 is the third device adapter in the segment or controller which contains it.
		
		Examples:
		<pre>6100 #5/Bus 0/3 (interface label omitted)
		5500 #3/Y-Bus/6/1 (second unnamed interface
		                          on adapter 6)</pre>
		Export always omits the interface name when there is only one device interface on the adapter or segment.

Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Interface Name".

Note: For those controller types which support a variable number of device interfaces, all controller device interfaces beyond the minimum number (which are created automatically) must be explicitly created with this record. If a CDI must be created, all intervening CDIs and device adapters will be created as necessary.

CU table

Table 20. CU Table

Field Name	Format	Description
Table Name	**Text**	**CU**
ID	Hex	<IODF Key> The ID of the IODF object.
Container	Text	Applicable only for new control units that are not already in the PWS configuration. Specifies the exact placement of the "new" control unit within a physical controller. A container object is either a controller label (for single segment controllers such as 3174), or a controller label/segment name, for controllers which have two segments. If not specified, HCM will place the device using the normal heuristics employed during the IODF load/resync process (this placement is based primarily on serial number). For example, 3174 #5, 3990 #12/SC0. Internal switch control units (those which are logically connected to port FE), and control units of type CTC or CFS, receive special treatment as follows: • Import ignores this field and the objects are placed automatically. • For CTC and CFS control units, export leaves this field blank since there is no meaningful container object for these control units. Ignored for control units which are already in the PWS configuration file (they have already been placed).
Serial number	Text EO	The serial number of the IODF object. **Note:** HCM uses the serial number of the container controller to update the IODF with control unit serial numbers.
Description	Text	<Updates the IODF> The description of the IODF object.
Type	Text EO	Value taken from the IODF object.

STRING table

Table 21. STRING Table

Field Name	Format	Description
Table Name	Text	**STRING**
Tag	Text	A unique tag to identify the string. This tag has no meaning outside the context of this import file.
		For new strings, any meaningful tag may be used. We recommend using the first device number in the string.
		For existing strings, HCM will first attempt to locate the one string which has this value as an explicit label. If no string was found, HCM will interpret the tag as a device ID and look for the string, labeled or not, that contains the device. **Usage Note:** unlabeled strings. If the string contains devices, the tag will be the device ID of the first (the lowest device number) of the string. If an unlabeled string has no devices, its tag will be of the form "String #*n*" where *n* is an integer value which is simply incremented each time such a tag is generated in order to guarantee uniqueness.

		Examples:
		 ```TSO POOL 1```
		 ```400```
		 ```400(MVS1)```

IODF device type	Text	The type-model of the string. For example, 3390.
		Applicable only for new strings that are not already in the PWS configuration. Ignored for strings which are already in the PWS configuration file.
		Export places the following in this field:
		The type-model of one of the device in the string with the lowest device number. If the string contains no devices, this field will be blank.
		See the "Usage Note" on page 347 in the PDF Name column.
PDF AbbrevId	Text	The abbreviated ID (AbbrevId =) of one of HCM's physical description templates. For example, RAMAC2.
		Applicable only for new strings that are not already in the PWS configuration. Ignored for strings which are already in the PWS configuration file.
		Export places the following in this field:
		If a specific PDF was used, and that PDF has an abbreviated ID, it will appear here. Otherwise this field will be blank.
		See the "Usage Note" on page 347 in the PDF Name column.

Table 21. STRING Table (continued)

Field Name	Format	Description
PDF Name	Text	The label (String =) of one of HCM's physical description templates. For example, IBM RAMAC 2 Array DASD.
		Applicable only for new strings that are not already in the PWS configuration. Ignored for strings which are already in the PWS configuration file.
		Export places the following in this field:
		If a specific PDF was used, the physical description name (String =) will appear here. Otherwise this field will be blank.
		Usage Note: Import usage of the ″IODF Device Type″, ″PDF AbbrevId″, and ″PDF Name″ fields:
		These fields are used to determine which HCM Physical Description should be associated with a new string. Only one of these fields is used on input. They are evaluated in the following order:
		1. PDF Name is used if the column is present and contains data. If specified it must be valid.
		2. PDF AbbrevId is used if the column is present and contains data. If specified it must be valid.
		3. HCM's normal search for a PDF based on the IODF Device Type field. If the IODF Device Type is blank or invalid, the default PDF is assigned.
Label	Text	A label for the string. Strings in HCM need not be explicitly assigned a label. When the label is absent, HCM will display a label of the form ″String *xxxx-yyyy*″ where *xxxx* is the device number of the first device in the string and *yyyy* is the device number of the last device number in the string. For example, string 300-31F.
Installed	Bool	Yes (this is the default) indicates that the string should be marked as installed.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

String interface (SI) table

Table 22. SI Table

Field Name	Format	Description
Table Name	**Text**	**SI**
Interface name	Text	The string tag followed by a slash "/" and the interface name. The interface name must be an allowable value in terms of the string PDF. For example, TSO POOL 1/A0, 400/A0, or 400(MVS1)/A0.
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Interface Name".

Note: All string interfaces beyond the minimum must be explicitly created with this record. If an SI must be created, all intervening SIs will be created as necessary.

UNIT table

Table 23. UNIT Table

Field Name	Format	Description
Table Name	**Text**	**UNIT**
Tag	Text	Identifier consisting of string tag followed by a slash "/" and a zero-based unit index. Units are numbered 0 to n-1 in row-major order. In strings which have the twoheaded attribute in the PDF, such as 4-path 3380, the units are counted from top to bottom in the left column, and then top to bottom in the right column. If the unit does not yet exist in the string, intervening units are created as necessary when creating the new unit. It is not possible to create additional units in the internal string of a switch.
Serial number	Text	The physical serial number of the unit.
Label	Text	The label of the unit.
Installed	Bool	Yes (this is the default) indicates that the string should be marked as installed.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

DEVICE table

Table 24. DEVICE Table

Field Name	Format	Description
Table Name	Text	**DEVICE**

ID	Text	<IODF Key> The ID of the IODF object. Normally the device number. To help ensure uniqueness, the device number is followed by the name, enclosed in parentheses, of one of the operating system configurations to which the device is defined. If the device is not defined to any operating system, nothing is added to the device number. On input, any of the defined operating systems will serve to uniquely identify the device. On output, export will choose the first defined operating system as sorted alphabetically in EBCDIC. Examples:
		
		`400`
		`400(MVS1)`

Container	Text	Applicable only for new devices that are not already in the PWS configuration. Specifies the exact placement of the device into a physical container. Container is a string tag or a unit tag. If a unit tag is specified, the device will be placed into that unit. If a string tag is specified, the device will be placed into the first available position in that string using HCM's normal heuristics. If this field is not specified, HCM will place the device using the normal heuristics employed during the IODF load/resync process. For example, 400, 400/0. On output, export always places a unit tag in this field. Internal switch devices (those which are logically connected via a control unit to port FE), and devices of type SCTC, BCTC, FCTC, CFS or CFP, receive special treatment as follows: • Import ignores this field and the objects are placed automatically. • For SCTC, BCTC, FCTC, CFS and CFP devices, export leaves this field blank since there is no meaningful container object for these devices. Ignored for devices which are already in the PWS configuration file (they have already been placed).
Serial number	Text	<Updates the IODF> The serial number of the IODF object.
Description	Text	<Updates the IODF> The description of the IODF object.
VOLSER	Text	<Updates the IODF> The VOLSER of the IODF object.
Type	Text EO	Value taken from the IODF object.

XBAR table

Table 25. XBAR Table

Field Name	Format	Description
Table Name	**Text**	**XBAR**
Label	Text	The physical label of the crossbar switch.
# of Channel Interfaces	Num	The number of interfaces across the top of the crossbar switch. For existing crossbars, decreasing the number of channel interfaces will cause connected channel interfaces to be disconnected before being deleted. This is an optional field. If a crossbar is being created, and this field is absent, the default number of channel interfaces (1) will be created.
# of Control Unit Interfaces	Num	The number of interfaces across the side of the crossbar switch. For existing crossbars, decreasing the number of control unit interfaces will cause connected control unit interfaces to be disconnected before being deleted. This is an optional field. If a crossbar is being created, and this field is absent, the default number of control unit interfaces (1) will be created.
Serial number	Text	The serial number of the crossbar switch.
Short name	Text	The short name of the crossbar switch.
Default XBARCFG	Text	Default crossbar switch configuration.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

XBARCFG table

Table 26. XBARCFG Table

Field Name	Format	Description
Table Name	**Text**	**XBARCFG**
Name	Text	The name of the crossbar configuration. This consists of the XBAR label followed by a slash "/" and the configuration name. The name of the default configuration "normal" is also accepted.
Row 1 Channel Interface	Text	The label of the channel interface to which this row (control unit interface) is connected. If this field is absent or blank, the row will be assigned no connectivity in this crossbar configuration. For example, 4.
Row 2 Channel Interface	Text	The label of the channel interface to which this row is connected. For example, 4.
...		
Row 32 Channel Interface	Text	The label of the channel interface to which this row is connected. For example, 4.

Crossbar channel interface (XCI) table

Table 27. XCI Table

Field Name	Format	Description
Table Name	**Text**	**XCI**
Interface name	Text	The XBAR label followed by a slash "/" and the interface name. The interface name is an integer from 1 to n where n is the number of channel interfaces.
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Interface Name"

Note: This record is exported, but ignored on import.

Crossbar control unit interface (XCUI) table

Table 28. XCUI table

Field Name	Format	Description
Table Name	**Text**	**XCUI**
Interface name	Text	The XBAR label followed by a slash "/" and the interface label. The interfaces are labeled A1, B2, C3, ..., Z26, AA27, AB28, ...
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Interface Name"

Note: This record is exported, but ignored on import.

CABINET table

Table 29. CABINET Table

Field Name	Format	Description
Table Name	**Text**	**CABINET**

Table 29. CABINET Table (continued)

Field Name	Format	Description
Id	Text	This, rather than the cabinet description, serves as the object identifier, since HCM enforces that it is unique.
Description	Text	The description of the cabinet.
Model	Text	The descriptive text describing the cabinet model. Any character string. Typically one of HCM's predefined cabinet models. For example, IBM Main Distribution Frame.
Serial number	Text	The serial number of the cabinet.
Graphical Position	Enum	One of HCM's values for "Suggested Graphical Position". Valid values are: no suggestion (this is the default), near processors, central, near devices.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

Note: The fields 'Naming format', '# of Panel Spaces', '# of Groups per Panel', '# of Ports per Group', 'Sequenced Row First' are no longer exported in the CABINET table, but they are accepted as input. If any of these fields are supplied, the import files will be treated as "old style".

PANEL table

Table 30. PANEL Table

Field Name	Format	Description
Table Name	**Text**	**PANEL**
ID	Text	This consists of the cabinet short name followed by a slash "/" and a two-character panel ID. If a panel of this ID does not exist in the cabinet, it will be added to the bottom of the cabinet.
Description	Text	The description of the panel.
Naming format	Text	The format string used to generate patchport labels for this panel. This must be specified and must be a valid format string if the panel is being created.
# of rows	Num	The number of rows of patchports in the panel. Used only if the panel is being created; ignored if the panel already exists. This field must be specified.
# of columns	Num	The number of columns of patchports in the panel. Used only if the panel is being created; ignored if the panel already exists. This field must be specified.

Table 30. PANEL Table (continued)

Field Name	Format	Description
Sequenced row first	Bool	If Yes, patchports are sequenced in a row-first manner when displayed in lists or used in operations involving multiple patchports.

PATCHPRT table

Table 31. PATCHPRT Table

Field Name	Format	Description
Table Name	**Text**	**PATCHPRT**
Name	Text	The name of the patchport as generated by the patchport naming format for this cabinet. Resolution of patchport names is made after all cabinet objects have been updated with the naming format values in the import file.
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Patchport Name".
Cabinet	Text EO	The short name of the cabinet which contains this patchport.
Explicit Label	Text	A name specified by you that is explicitly assigned to the patchport (overriding the name which would be assigned with the patchport naming format).
Installed	Bool	Yes, if the group containing this patchport should be marked as installed. There is no default for this field, when no data is present here, the installed status is left unchanged. When panel spaces are created, all groups are marked as installed. Since patchports are installed in groups, export will place the same value in this column for all patchports in the group. On import, all patchports in the group are expected to have the same value for this attribute, but this is not strictly enforced. If the specified value does not match that of other PATCHPRT records in the import file having the same group, the last PATCHPRT record found in the import file will win. You will *not* be informed with a warning message.

CONVERTR table

Table 32. CONVERTR Table

Field Name	Format	Description
Table Name	**Text**	**CONVERTR**
Label	Text	The label of the converter. If this label is not unique in the configuration, the row cannot be processed. **Note:** Export will assign names to generic converters (converters which have no label). Importing this file will result in these labels being assigned to the converters. Export will assign names in the following format: • S->P #*nnn* • P->S #*nnn* This is an instance where an export operation followed by an import operation will not recreate the original configuration exactly.
Unique Key	Text EO	The unique key for this object. Composed of "Table Name:Converter Label".
Type	Enum	One of HCM's allowed converter types, as they appear in the Type: combo box of the Create Converter dialog. Valid values are: parallel->serial, serial->parallel.
Position	Enum	One of HCM's allowed values for position. Valid values are: top, bottom. This value is not used for unused converters (for example, for ones which are not connected). Export places the value "top" in this field for unused converters.
Serial number	Text	The serial number of the converter.
User field 1	Text	User field 1
User field 2	Text	User field 2
User field 3	Text	User field 3
...	...	...
User field 20	Text	User field 20

CABLE table

Table 33. CABLE Table

Field Name	Format	Description
Table Name	Text	CABLE
ID	Text	The cable identifier.
Type	Enum	The three-character abbreviation for one of HCM's allowed cable types:
		GFP Gray Fully Populated
		GPP Gray Partially Populated
		BFP Blue Fully Populated
		BPP Blue Partially Populated
		BDI Bi-directional
		SMJ Single Mode Jumper (fiber)
		MMJ Multi Mode Jumper (fiber)
		SMT Single Mode Trunk (fiber)
		MMT Multi Mode Trunk (fiber)
		This field is required when creating a new cable.
		Note: If the type of an existing cable is changed to be incompatible with the interfaces it connects with, the cable will be removed from the connection and placed into the pool of "Unused cables". The interfaces will remain connected with a generic cable.
Length	Text	A free form character string describing the length. For example, 50m.
Loss A	Float	The value for dB Loss A of this fiber cable, in dB.
		For cables which are not composed of fiber optic: • Import ignores the value. • Export places 0 in this field.
		On import, the value 0 is treated as "N/A" (not available) by HCM. This is the default for new cables.
Loss B	Float	The value for dB Loss B of this fiber cable, in dB. Same considerations as Loss A.
Comment	Text	User defined comments and text.

LOGLINK table

Table 34. LOGLINK Table

Field Name	Format	Description
Table Name	Text	**LOGLINK**
Source	Text	The name of the object which begins this logically complete series. For example, PROC1/30.
Source Type	Enum	Object type qualifier (table name) for the source. This is only necessary if the source does not uniquely identify a connectable object (for example, it is legal to have a processor named A1 and a switch also named A1, so A1/D2 could be a CHPID or a PORT; in such a case source type must be supplied to distinguish which of the two objects is being connected). If it is supplied, source must identify an object of the given type.
		If omitted, the source is assumed to be unique and the object type will be inferred.
		Valid values are: CHPID, PORT, XCUI, CDI, CONVERTR, PATCHPRT.
Source Unique Key	Text EO	The unique key for the source object. Composed of "Table Name:Object Name".
Target	Text	The name of the object which ends this logically complete series. For example, 3990_#12/SC0/A.
Target Type	Enum	Object type qualifier for target. Follows the same rules as source type.
		Valid values are: CCI, XCI, SI, PORT, CHPID, CONVERTR, PATCHPRT
Target Unique Key	Text EO	The unique key for the target object. Composed of "Table Name:Object Name".
Transmit Loss	Float	The transmit loss of the entire fiber link, in dB.
		For series which are not composed of fiber cables: • Import ignores the value. • Export places 0 in this field.
		On import, the value 0 is treated as "N/A" (not available) by HCM.
Receive Loss	Float	The receive loss of the entire fiber link, in dB. Same considerations as transmit loss.

Note: This table is used in two ways:
1. On import, its only purpose is to update the transmit and receive loss values for a fiber link.
2. On export, each LOGLINK row serves as an anchor for one or more rows of the CONNECT table.

Each LOGLINK row represents a "Logically Complete" series

A series of connections is logically complete if it starts at a logical connection source and ends at a logical connection target. For fiber connections, this is usually equivalent to a fiber link. For example, the sequence of connections from PROC1, CHPID 30 to patchport A1C6, then from patchport A1C6 to 3990 #2/SC0/A is a logically complete series from PROC1/30 to 3990 #2/SC0/A.

The source and target columns identify the end points of a logically complete series.

Valid sources for a "Logically Complete" series: Think of a LOGLINK source as the electronics which send signals, the originator of the signal on this cable:

  CHPIDs

  PORTs (Switch ports, when used as control unit ports, or ports connecting to chained switches)

  XCUIs (Crossbar control unit interfaces)

  CDIs (Controller device interfaces)

  CONVERTRs (the "bottom" of the converter. Both sides of the converter must be connected.)

  PATCHPRTs (only when permanently connected on one side, and not connected on the other)

Valid targets: Think of a LOGLINK target as the electronics which receive signals, the eventual destination:

  CCIs (Controller channel interfaces)

  XCIs (Crossbar channel interfaces)

  SIs (String interfaces)

  PORTs (Switch ports, when used as CHPID ports, or ports connecting to chained switches)

  CHPIDs (when used in point to point CTC connections and Coupling Facility connections)

  CONVERTRs (the "top" of the converter. Both sides of the converter must be connected.)

  PATCHPRTs (only when permanently connected on one side, and not connected on the other)

Note: Not all combinations are possible or meaningful. For example, it is not possible to create a series between a string interface and a CHPID.

Special considerations for daisy-chained connections: Where daisy-chained connections are involved, the logically complete series will be between the source and the final destination at the end of the daisy chain. For example, a connection from CHPID 01 to 3172 #1 then daisy-chained to 3172 #2 would be represented by a LOGLINK between CHPID 01 and 3172 #2.

Special considerations for converters: Each converter in a chain of connections starts a new logically complete series. Thus, any LOGLINK that has a parallel source will have a parallel target. Any LOGLINK that has a serial source will have a serial target. Breaking up series based on converter positions is useful for two reasons:
1. It allows a LOGLINK which specifies transmit loss and receive loss values to associate those losses just with the serial portion of a chain of connections.
2. That fact that a chain of connections may "branch" at a parallel to serial (P->S) converter due to daisy chaining.

Consider the following example, in which a connection "branches" at a P->S converter such that two controller interfaces are reachable by one CHPID:

```
CHPID 01 ==== >  /In P->S Converter ----> Serial Controller Interface (A)
          !== <  /Out
          !
          !== >  /In Parallel Controller Interface (B)
```

Read the above example as a sideways pictorial diagram. Double lines "===" are parallel, single lines "---" are serial.

In this case, there will be two logically complete series, and two LOGLINK records:
> From CHPID 01 to Interface B (because for daisy chains the target is the "final" one); and
> from the P->S Converter to Interface A.

As a further illustration, consider how many LOGLINK records would be needed if the daisy chaining were not to a parallel controller interface, but rather to another serial controller interface (via another P->S converter):

```
CHPID 01 ==== >  /In P->S Converter ----> Serial Controller Interface (A)
          !== <  /Out
          !
          !== >  /In P->S Converter ----> Serial Controller Interface (C)
```

In this case, there will be three LOGLINK records:
> From CHPID 01 to the second P->S Converter.
> From the first P->S Converter to Interface A.
> From the second P->S Converter to Interface C.

Order of processing: LOGLINK table is processed after all other records have been processed, regardless of their order in the import data stream. This is because the validity of the series depends on the existence and type of connections, some of which may not be mentioned until later in the import file.

Connections from CHPIDs on OPPP processors

When dealing with OPPP (One Processor Per Partition) style processors, it is possible to have more than one CHPID (from different logical processors) correspond to the same physical channel path number. That physical channel path may be connected to something. In that case, export will only produce one LOGLINK record for that path. Consider a case in which two CHPIDs map to the same path. The single, exported LOGLINK record will contain the CHPID source identification of one of the two CHPIDs. No other LOGLINK record is required because the connectivity of the second CHPID can be inferred from the connectivity of the first, since the physical manifestation as a path is shared.

If you are creating files for import you should use the same philosophy as that used by export, and avoid entering more than one LOGLINK record per path.

The same applies to CONNECT records, described below. That is, for each path, there should be only one CONNECT record that lists a CHPID on that path as its source.

CONNECT table

Table 35. CONNECT Table

Field Name	Format	Description
Table Name	**Text**	**CONNECT**
Source	Text	The name of the source of this connection. This field is composed of the object name, followed by an optional qualifier for those objects such as patchports which have two interfaces (front and back). See "Source and target optional qualifiers" on page 360
Source Type	Enum	Object type qualifier (table name) for the source. This is only necessary if the source does not uniquely identify a connectable object (for example, it is legal to have a processor named A1 and an ESCON switch also named A1, so A1/D2 could be a CHPID or a PORT; in such a case source type must be supplied to distinguish which of the two objects is being connected). If it is supplied, source must identify an object of the given type.
		If omitted, the source is assumed to be unique and the object type will be inferred.
		Valid values are: CHPID, PORT, XCUI, CDI, CONVERTR, PATCHPRT. Also valid (for daisy-chaining support) are: CCI, SI, XCI.
Source Unique Key	Text EO	The unique key of the source object. Composed of "Table Name:Object Name".
Target	Text	The name of the target of this connection. Follows the same rules as Source.
Target Type	Enum	Object type qualifier (table name) for the target. Follows the same rules as source type.
		Valid values are: CCI, XCI, SI, PORT, CHPID, CONVERTR, PATCHPRT.
Target Unique Key	Text EO	The unique key of the target object. Composed of "Table Name:Object Name".
LOGLINK Target Unique Key	Text EO	The unique key of the target object in the logically complete series which contains this connection. All connections in a logically complete series have the same value for this field.
		This field, along with the sequence field, enables a report program to collect the individual connections in a logically complete series.
Sequence	Num EO	The sequence of this connection in the logically complete series. Connections are numbered from the bottom up (from the target to the source), starting with 1.

Table 35. CONNECT Table (continued)

Field Name	Format	Description
Cable ID	Text	The cable ID to be used for this connection. The specified cable must exist in the configuration, and its cable type must be compatible with this connection. If the cable is already used in another connection, it will be reassigned to this connection. The previous connection will not be broken. If this field is omitted, mutable connections will be made without a named cable, and permanent connections will be made with an anonymous trunk.
Permanent	Bool	Yes will mark this connection as permanent.

Note: The sequence field is numbered from the target towards to source. It could have been done the other way around. The target to source sequence was chosen since in HCM, you always connect targets to sources, and usually view connections (fiber links) from the target's perspective. To maintain consistency with HCM base, the sequence starts at the target and looks upwards.

Each row in the CONNECT table represents a discrete connection between two adjacent objects, a connection which can be described with a single cable. A logically complete series (LOGLINK) is often composed of several CONNECTS. For example, the sequence of connections from PROC1, CHPID 30 to patchport A1C6, then from patchport A1C6 to 3990 #2/SC0/A requires two CONNECT rows:

> PROC1/30 to patchport A1C6/back, and
> patchport A1C6/front to 3990 #2/SC0/A

The LOGLINK table serves as a means to associate these two connection records. The LOGLINK table, with the sequence field of the CONNECT rows, is sufficient to report on the complete sequence of connections from the original source to the eventual target.

The source and target columns identify the end points (interfaces) of the connection. Before making the connection, HCM will remove current connections from the objects if necessary.

Source and target optional qualifiers

Some objects have more than one possible cable connection. In these cases, an additional qualifier is necessary to precisely indicate the interface. These qualifiers are:

- For patchports:
  > /front (or just /f)
  > /back (or just /b)
  > The /front or /back qualifier is always required for CONNECTs involving PATCHPRTs.
- For interfaces which can be daisy-chained, including parallel interfaces and string interfaces:
  > /in
  > /out
  > If this qualifier is absent during import, HCM assumes /in.

Notes:

1. Including /in and /out makes it easier to understand reports generated from export data.
2. The slash "/" used in the examples above is meant to represent the concatenation character, for which you can select a different symbol.

Valid sources and targets

Sources and targets are any interfaces that can be connected to HCM. The following are valid sources or targets:
- CHPIDs
- PORTs
- XCUIs (Crossbar control unit interfaces)
- CDIs (Controller device interfaces)
- PATCHPRTs (always qualified with /front or /back)

The following are daisy-chained connections proceeding onwards from a parallel interface:
- XCIs (Crossbar channel interfaces, optionally qualified with /out)
- CONVERTRs (may be qualified with /in or /out if it is the top of a parallel->serial converter)
- CCIs (Controller channel interfaces; if they are bus-and-tag, they may optionally be qualified with /in or /out)
- SIs (String interfaces, optionally qualified with /out)

If you use OPPP (One Processor Per Partition) style processors, you should only create one CONNECT record per physical channel path in your import files, even if the path is shared by more than one CHPID. You can find more information in the "LOGLINK table" on page 356 under "Connections from CHPIDs on OPPP processors" on page 358. Export automatically restricts its output so that there is only one CONNECT record per physical channel path.

Regardless of the type of connection, the following general rules apply:
- An object may not be connected to itself.
- The interface types (parallel, serial, ...) of the source and target must match.
- A connection source, or target, with explicit or defaulted qualifiers, may appear only once in the CONNECT table.
- A cable ID may appear only once in the CONNECT table.
- Converter connections are only honored if they complete a connection from a signal source (e.g. CHPID or PORT) to a receiver (e.g. XCI or CCI).

Connections involving ports on switches

Physical connections involving ports on switches are present in both the PWS file and the IODF. For every physical connection or disconnection caused by the import file, appropriate changes will also be recorded in the IODF. HCM will update the Port<>CU, Port<>CHPID, and Port<>Port links in the IODF in a similar manner to the actions which would have been taken if you had connected or disconnected using the standard HCM user interface.

Note: All IODF updates are confirmed by you. See "General Concept on IODF Update Processing" on page 324 for more information.

Order of processing

The CONNECT table is processed before the LOGLINK table, and after all other tables have been processed, regardless of their order in the import file. This is because connections depend on the existence and type of interfaces and cables, some of which may not be mentioned until later in the import file.

PPRCLINK table

Table 36. PPRCLINK Table

Field Name	Format	Description
Table Name	Text	**PPRCLINK**
Source Interface	Text	The interface of the primary subsystem that is involved in the PPRC link.
Source Control Unit	Hex	The control unit of the primary subsystem that is involved in the PPRC link.
Target Control Unit	Hex	The control unit of the secondary subsystem that is involved in the PPRC link.
Link Address	Text	The two-digit link address used for the PPRC connection if the connection involves a switch.
CUADD	Text	The CUADD value (two-digit hexadecimal number) used for the PPRC connection.
Description	Text	The description of the PPRC link.
DEVN	Text	Four-digit hexadecimal number indicating the target device of CESTPATH commands, also provided by the user when creating or editing a PPRC connection. This should be a device within the primary subsystem.
Primary SSID	Text	The subsystem ID of the primary subsystem.
Secondary SSID	Text	The subsystem ID of the secondary subsystem.
Link	Text	This 8-digit hexadecimal number has a format borrowed from host-based PPRC tools: 'SSSSLLCC' where SSSS is the four-digit System Adapter Identifier (SAID), LL is the two digit link address, and CC is the two-digit destination LCU (i.e. CUADD).

DEVCU table (export-only)

Table 37. DEVCU Table (export-only)

Field Name	Format	Description
Table Name	Text	**DEVCU**
Device ID	Text EO	<IODF Key> The ID of the IODF object. Normally the device number, but may be further qualified as described for the ID field of the DEVICE table.
CU ID	Hex EO	<IODF Key> The ID of the IODF object.

Note: This table is provided to facilitate the use of export files to create reports on logical connectivity of devices to control units.

CUCHPID table (export-only)

Table 38. CUCHPID Table (export-only)

Field Name	Format	Description
Table Name	Text	**CUCHPID**
CU ID	Hex EO	<IODF Key> The ID of the IODF object.
CHP ID	Text EO	<IODF Key> The ID of the IODF object, formatted as described in the CHPID table for the ID field.

Table 38. CUCHPID Table (export-only) (continued)

Field Name	Format	Description
DLA	Hex EO	Dynamic link address, valid only if the DLA Set field evaluates to a Yes.
DLA Set	Bool EO	Yes if there is a dynamic link address (DLA).

Note: This table is provided to facilitate the use of export files to create reports on logical connectivity of control units to CHPIDs.

DEVIMAGE table (export-only)

Table 39. DEVIMAGE Table (export-only)

Field Name	Format	Description
Table Name	**Text**	**DEVIMAGE**
Device ID	Text EO	<IODF Key> The ID of the IODF object. Normally the device number, but may be further qualified as described for the ID field of the DEVICE table.
Image ID	Text EO	<IODF Key> For images on BASIC mode processors, consists of a processor ID (for example, PROC1). For images on LPAR mode processors, consists of a processor ID, a separator, and a logical partition name (for example, PROC1/LPAR1). The presence or absence of a separator in this field can thus be used as an indicator of what type of image this is.
Link type	Enum EO	Valid values are: **A** Device reaches image via a CHPID that has that image in its access list. **C** Device reaches image via a CHPID that has that image in its candidate list. **-** Device reaches image via a CHPID that has that image in either its access or candidate list, but the image is not in the device's explicit candidate list. **+** Device does not connect to any CHPID that has this partition in its access or candidate list, yet the device's explicit candidate list includes that image.

Note: This table is provided to facilitate the use of export files to create reports on logical connectivity of devices to images (taking into account any explicit candidate lists). Normally, this table can be exported regardless of whether you are running in standalone or online mode. However, if the configuration has devices which are connected to a processor via a reconfigurable (REC) CHPID, communication with the host is required in order to determine whether the processor model is EMIF capable or not. In online mode, the communication takes place automatically and transparently. In standalone mode, you will see an appropriate error message indicating that the attempted operation is not permitted in standalone mode. The name of the table currently being exported appears in the status bar, so you will know which table required host access.

Chapter 16. Providing input to the Tivoli Inventory database

Tivoli Inventory is a hardware and software inventory gathering application for the Tivoli Framework, designed to help system administrators and accounting personal manage their client/server environment. With Tivoli Inventory it is possible to gather and maintain hardware and software information for every node in a Tivoli Management Region. HCM provides optional Tivoli Inventorysupport by making HCM exported data accessible to Tivoli Inventory. In particular, HCM exported data can be converted into a format that can be read by Tivoli Inventory scans, and thus HCM data can be maintained in the Tivoli Inventory database.

For information on installation prerequisites, refer to "Prerequisites" on page 3.

Updating the Tivoli Inventory database definition for HCM exported data

During the HCM installation (see Chapter 2, "Installing HCM," on page 3), the following files for Tivoli Inventory support have been copied to the ...\MIFUTIL directory. MIFUTIL is a sub directory of the installation directory you specified during the installation process. The default for the installation directory is \IBMEEQ. The files which belong to the Tivoli Inventory support are listed in Table 40.

Table 40. Files for Tivoli Support

[Drive]:\IBMEEQ\MIFUTIL\EEQ2MIF.EXE	Conversion tool
[Drive]:\IBMEEQ\MIFUTIL\EEQ2TMEM.SQL	SQL script for MS SQL
[Drive]:\IBMEEQ\MIFUTIL\EEQ2TMES.SQL	SQL script for Sybase
[Drive]:\IBMEEQ\MIFUTIL\EEQ2TMEO.SQL	SQL script for Oracle
[Drive]:\IBMEEQ\MIFUTIL\EEQ2TMED.SQL	SQL script for DB2
[Drive]:\IBMEEQ\MIFUTIL\TABLDESC.TXT	Tables that store HCM exported data

The database tables predefined by Tivoli Inventory are not designed to hold host hardware configuration data. Run the EEQ2TMEx.SQL script suitable for the used RDBMS to extend the Tivoli Inventory database schema with host hardware configuration data. Perform the following steps:

1. Copy the appropriate SQL script to your database server.
2. Log on to the inventory database as database owner or database administrator.
3. Invoke the SQL script. The script creates all necessary tables for the HCM exported data. For a description of these tables and its columns please refer to "Table description" on page 372. How to invoke the SQL script depends on the database system and the database tools you have installed. See the *Tivoli Inventory 3.2 User's Guide* for more information.

The basic database configuration that is created by Tivoli's SQL scripts is a minimal configuration. In most cases, the database size will be too small to hold large hardware configurations exported by HCM. Ask your database administrator to adjust the sizes according to the amount of data you want to store. Since large

hardware configurations will also cause large database transactions to populate the appropriate database tables, you additionally must increase the size of the rollback spaces.

The performance of HCM Tivoli Inventory support mainly depends on the performance of the used database when processing large transactions. Therefore, it is strongly recommended to choose appropriate database parameters to get the best possible performance. See your database documentation and ask your system administrator.

Customizing a Tivoli Inventoryprofile for custom MIF file scan

Tivoli Inventory gathers information about HCM data by scanning MIF files. The conversion tool creates these files by converting the HCM exported files to MIF format. You have to create and then customize a Tivoli Inventory profile which starts the conversion tool on your HCM workstation and scans the produced MIF files. To create a Tivoli Inventory profile, start the Tivoli desktop on the TMR server, open the appropriate policy region and profile manager, and choose **Create->Profile...** from the menu in the profile manager window. Select **InventoryProfile** as profile type. For more information about customizing a Tivoli Inventory profile see the *Tivoli Inventory 3.2 User's Guide*. The following figure shows a sample of the **Customize Inventory Profile** window.
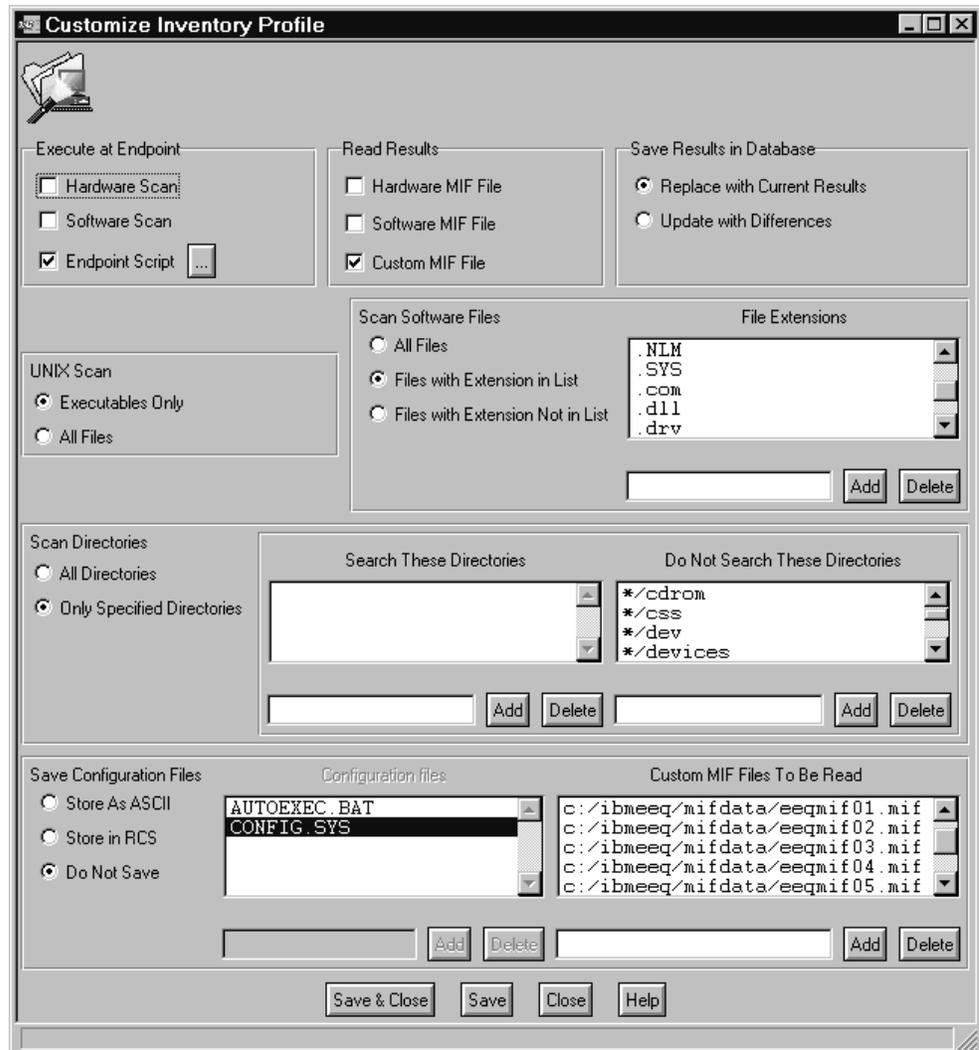
Figure 159. Customize Tivoli Inventory Profile

You have to make the following settings:

- Inventory must start the conversion tool at the HCM workstation. Therefore, an endpoint script must be executed. In the **Execute at Endpoint** box select **Endpoint Script**. The endpoint script must contain the following statement:

```
[Drive]:\IBMEEQ\MIFUTIL\EEQ2MIF [Option][String]
```

IBMEEQ is the default installation directory for HCM. If you have chosen a different directory during installation you must replace IBMEEQ accordingly. See "Invoking the conversion tool" on page 371 for an explanation of the [Option] and [String] parameter.

Switch off **Hardware Scan** and **Software Scan**.

- If you want to submit any other commands you can include them in the endpoint script as well. For example, you can run the following commands before starting the conversion tool to make sure that all obsolete MIF files are deleted before creating the new files.

```
attrib -R [Drive]:\IBMEEQ\MIFDATA\*.MIF
del [Drive]:\IBMEEQ\MIFDATA\*.MIF
```

The endpoint script is a batch file which is executed at the endpoint. For further information see the *Tivoli Inventory 3.2 User's Guide*.

- The following figure shows an example of an endpoint script:
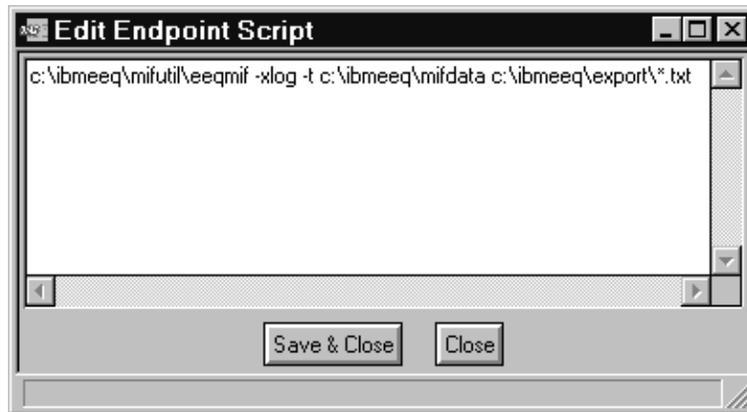
Figure 160. Example of Endpoint Script

- In the **Read Result** box choose **Custom MIF File**. Switch off the **Hardware MIF File** and **Software MIF File**.
- See the *Tivoli Inventory 3.2 User's Guide* for the appropriate setting of the radio buttons in the **Save Results in Database** box.
- As you only scan a custom MIF file, the group boxes **UNIX Scan**, **Scan Software Files** and **Scan Directories** are not relevant.
- In the **Save Configuration Files** choose **Do Not Save**.
- In the **Custom MIF File to be Read** box add the absolute path to your target directory and the file names of the MIF files to be scanned. The MIF files are named EEQMIFxx.MIF, where xx are the two digit hexadecimal numbers from 00 to 19.
  - [Drive]:\IBMEEQ\MIFDATA\EEQMIFxx.MIF

  [Drive]:\IBMEEQ\MIFDATA is the default target directory for the MIF files produced by the conversion tool. If you have specified a different directory in the endpoint script (eeq2mif.exe invocation) you must replace the default path here.
- Click **Save & Close** to save your profile and leave the profile definition dialog.

To distribute the customized profile, click on **Profile Manager** from the menu bar and select **Distribute** from the pull down menu, or just drag & drop the profile icon to the subscriber icon. The subscriber to where you distribute your profile must be the HCM workstation. If your HCM workstation is not already a subscriber to the profile you can subscribe it by dragging its icon from the policy region to the profile manager's subscribers area. When running the scan, the endpoint script will be executed which invokes EEQ2MIF.EXE. EEQ2MIF will convert the specified HCM exported files to MIF file format (Before scanning, data has to be exported by HCM, see "Exporting data from HCM for Tivoli Inventory support" on page 369). Tivoli Inventory reads these MIF files and the data is placed in the Tivoli Inventory database in the corresponding database tables. Tivoli Inventory queries can be added to an existing or a new query library. These queries can then be used to retrieve certain data from the HCM tables.

There are several ways to check if the scan has run through:

- Check the conversion tool log file for any error or warning messages that have been issued. The log file is located in the target directory.
- Check the Tivoli Inventory notice group for messages.
- The column CONFIG_CHANGE_TIME in all HCM tables in the inventory database is an indicator when the last scan has written data to the database.

If an error occurs during the scan process, this information can be used to determine which component caused the problem.

Because the workstation ID is by default a primary key for the HCM tables in the Tivoli Inventory database there can be only one set of HCM exported data for each HCM workstation. With each new scan the HCM data in the Tivoli Inventory database is replaced by the last exported HCM data. The exact behavior on how Tivoli Inventory handles results from scans depends on the **Save Result in Database** selection in the Tivoli Inventory Profile. Refer to the *Tivoli Inventory 3.2 User's Guide* for more information.

Creating a query

In order to view the information Tivoli Inventory has scanned, customize a Tivoli Inventory query. Perform the following steps:

1. If you have not already created a query library, do the following: From the Tivoli Inventory desktop open the Policy Region in which you want to create the QueryLibrary and choose **Create->QueryLibrary** from the menu.
2. Open the query library and choose **Create->Query** from the menu.
3. In the Edit query dialog type in the name of the HCM table from which you want to retrieve information. Select one or more columns from the column list. For an overview of the HCM table definitions see "Table description" on page 372. To specify the data more precisely you can also enter a "where" clause.
4. Once you have customized the query to your needs you can save and run it.

Tivoli Inventory will display the retrieved data in a result window.

Exporting data from HCM for Tivoli Inventory support

The conversion tool EEQ2MIF.EXE converts HCM exported data to MIF file format. It is necessary to export your HCM data before starting a custom MIF file scan each time your configuration has changed. The directory to which you export your data must be the same you have specified as source directory for the conversion tool. Export your HCM configuration data as follows:

Start HCM on your workstation and open the IODF from which you want to export data.

From the File menu choose **File->Export Data...**
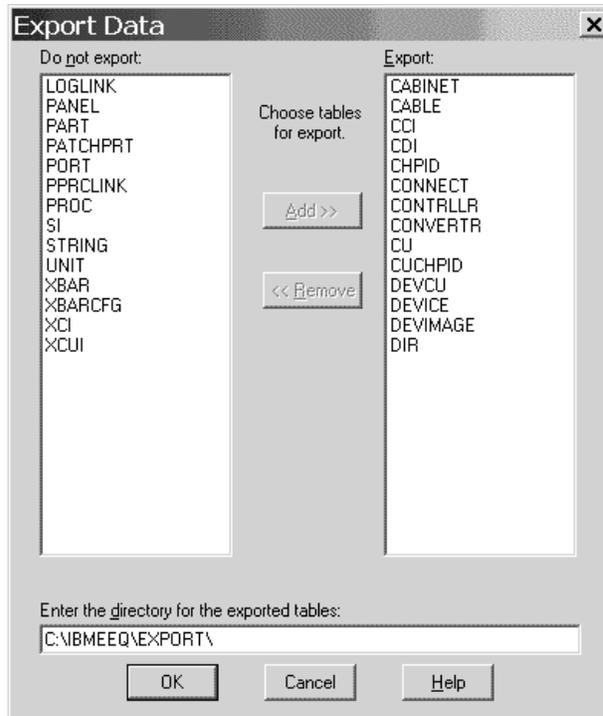
Figure 161. Export Data Dialog

Add the tables you want to export to the Export list. Enter a valid path for your exported data files and click **OK**. The path must be the same as the specified source directory for the conversion tool.
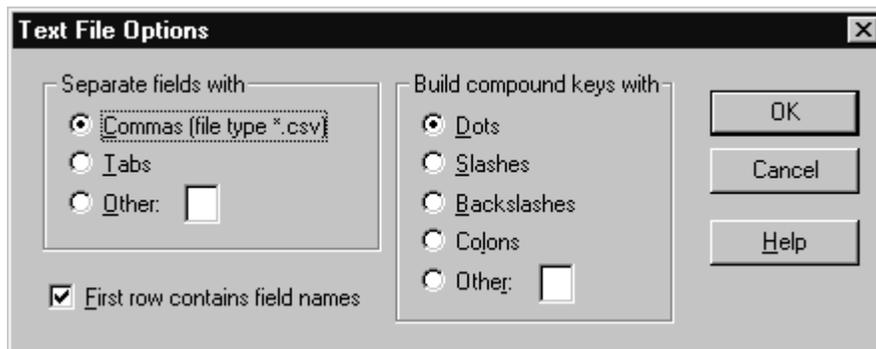
Figure 162. Text File Options Dialog

Only commas as delimiters are supported by the conversion tool. You must choose **Comma** in the **Separate fields with** box. As the conversion tool needs the field names as input you must select the **First row contains field names** box. The **Build compound keys with** box is not relevant. Click the **OK** button to confirm your choices.

HCM now exports your hardware configuration data into the specified directory. For each table a separate file is created. For a detailed description of the file and tables format see Chapter 15, "Importing and exporting data," on page 317.

> **Tip**
>
> It is not necessary to export all available tables. You can omit all tables in
> which you are not interested. It is recommended to omit the tables
> CUCHPID, DEVCU, and DEVIMAGE. If you want to store large hardware
> configurations, the performance is significantly improved when you omit
> these tables.

Invoking the conversion tool

The endpoint script must contain a command to invoke the conversion tool. The
syntax of this command has the following form:

```
EEQ2MIF [Option] [String]
```

EEQ2MIF.EXE converts information from the specified HCM exported files into a
set of MIF files. If all specified source files can be converted without errors, the
files EEQMIFxx.MIF (with xx in the range of hexadecimal numbers from 00 to 19)
will be created in the target directory. Each MIF file corresponds to one HCM
exported file. If an HCM exported file is empty or not available (in case of
wildcard usage), the respective MIF file will be created but will contain no data
records.

If an error occurs during conversion, the MIF files EEQMIFxx.MIF in the target
directory will be renamed to EEQMIFxx.MI~. The new extension prevents Tivoli
Inventory from reading inconsistent MIF files. Tivoli Inventory will stop the
processing of the distributed profile and report that some MIF files cannot be read.
Take a look into the log file of the conversion tool to check the messages.

Option

[Option] is an optional parameter and refers to the available options the routine
can be started with.

These options concern the following:
* The building of a log file EEQ2MIF.LOG containing status and diagnostic
  messages which may occur during the conversion process
* The specification of a target directory for the produced MIF files and log file
* An available help function to show all possible options which can be specified

Available options are:

[Option] = -xlog   An extended version of a log file is produced

[Option] = -t [Target]   This option specifies a target directory [Target] the
produced MIF files and log file are to be written to.
The default target directory is ..\MIFDATA\ relative
to the directory in which HCM resides. If no target
option is given, the MIF and log file will be written
to this default directory.

The target directory [Target] has to be specified in
the following manner:

```
[Target] = [Drive]:\[Dir]\...\...\
```

where [Drive] denotes a valid drive letter and [Dir] a directory name. You can omit the ending backslash \.

If the target directory does not exist it will be created.

[Option] = -? , -help

This option invokes a help function which displays a help message on the screen. Use this option only if you invoke the conversion tool manually.

If the -help or -? option is stated a possibly following [String] statement is not considered.

You can also invoke these options with /[Option] instead of -[Option].

String

[String] is a required parameter and locates the HCM exported files to be read. You can define [String] in the following ways:

- [String] = [Drive]:\[Dir]\...\...\

  All files in this directory will be processed

- [String] = [Drive]:\[Dir]\...\[File]

  The file [File] in the specified directory will be processed.[File] may contain wildcards (see below).

- [String] = [File]

  The file [File] in the current directory will be processed.

  [File] may contain wildcards in the following way:

  [File] = *.[Ext]

  where [Ext] denotes any file extension.

Table description

The tables that were added to the inventory database in order to store the HCM exported data are described in Appendix F, "Table description for Tivoli Inventory Support," on page 421. The tables are also stored in the product directory [Drive]:\IBMEEQ\MIFUTIL\TABLDESC.TXT.

Chapter 17. Physical description files (PDFs)

Controllers and strings come in various physical arrangements. In some cases there are alternate physical arrangements available for the same unit type. HCM uses *physical description files* (PDFs) to represent the non-default physical arrangements of controllers and strings. A default controller is one that has one or more channel interfaces, one segment, and one device interface. It cannot be grouped with other controllers. A default string is one that has one interface and eight devices per unit.

A PDF is an ASCII text file containing one or more (usually one) physical descriptions for controllers or strings. The physical description is a template of an object hierarchy that can be used to characterize a controller (or set of controllers) or a string.

Note: The filename extension for physical description files was changed from ″.pdf″ to ″.txt″ for HCM 2.9 Service Level 3 and higher. Nevertheless, HCM 2.9 Service Level 3 and higher can still read physical description files having the extension ″.pdf″.

Loading PDFs

PDFs are stored on the PWS disk where HCM is installed. Controller PDFs are in the following directories: CPDFA, CPDFB, CPDFC. String PDFs are in the following directories: SPDFA, SPDFB, SPDFC. The prefix 'C' or 'S' designates controller or string. The final 'A', 'B', or 'C' determines the precedence given to PDFs during an IODF load or resync (for more details, see "How PDFs are assigned to objects" on page 71). HCM loads all PDFs from these directories when a new configuration is created or an existing one is opened.

Customizing PDFs

PDFs can be changed and new PDFs can be added to the appropriate directories as needed. Changes to PDFs do not take effect until the next time an HCM configuration is created or opened. If a PDF for an existing controller or string is changed (e.g. the maximum number of channel interfaces on a controller segment is increased or decreased), then any further changes to the physical arrangement of that piece of equipment will be constrained by the new PDF.

Notes:

1. Changing a PDF never causes physical changes to a controller or string automatically.
2. The easiest way to create your own PDF is to copy an existing PDF that is close to your needs. Then modify the copied PDF according to your needs, for example, adjust the number of segments or interfaces. Save your modified PDF file for example in one of the directories CPDFA or SPDFA, depending on whether you have modified a controller or string PDF. Load your HCM configuration file, and apply now the new PDF to all objects to which you want to apply it.
3. Do not store your private changed or modified PDFs in the 'B' directory. An installation of a new HCM version or service level might reset all your private changes and activities of PDFs.

PDF syntax

In PDFs, an exclamation mark (!) indicates the beginning of a comment; parsing resumes on the next line. Indentation is recommended but not necessary for parsing. Recognition of keywords is not case-sensitive (i.e. they may be in mixed upper or lowercase). Labels will be stored as they appear in the PDF (the parser does not convert labels to upper or lowercase). Labels can contain spaces if they are enclosed in double quotes.

Brackets [ ] indicate an optional clause.

Braces { } indicate a choice of entry; unless a default is indicated, you must select one of the entries.

Items separated by a vertical bar | represent alternative items. Select one of these items.

An ellipsis (...) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.

Parentheses (not-bolded) are used to enclose clauses of which multiple entries are allowed.

CAPITAL LETTERS (including punctuation) indicates the exact characters to be entered. Such items must be entered exactly as illustrated, except that letters can be upper case or lower case.

CAPITAL LETTERS UNDERSCORED indicates a default value that is used if you omit the clause.

Controller PDF syntax

A controller PDF is an ASCII text file that contains one or more controller physical descriptions. A controller physical description has the following syntax.

```
      SUBSYSTEM = label
      [ABBREVID = id]
      UNITTYPE = unit
         CLASS = { DASD | TAPE | TERMINAL | "UNIT RECORD" |
            TELECOMMUNICATIONS | MICR/OCR | GRAPHICS |
            OTHER | "ESCON CTC" ]
         [COMGROUP = {TRUE | FALSE}]
         [CANGROUPADAPTERS = {TRUE | FALSE}]
         [OWNSALLDEVICES = {TRUE | FALSE}]
         [INTEGRATED = {TRUE | FALSE}]
      ( CONTROLLER [ = label ]
            [ [ALLOW {number | (number[,number]...)}]
      [DEFAULT number]
      SEGMENTS
            ]

      ( SEGMENT [ = label ]
      [ALLOW {number | (number[,number]...)}]
      [DEFAULT number]
      CUINTS = ( label [[number]] [,label [[number]]]... )

      [ALLOW {number | (number[,number]...)}]
      [DEFAULT number]
      DEVINTS = ( label (,label)... )
            )...
      )...
```

For controllers that have multiple channel interfaces per channel adapter, the syntax of the SEGMENT clause is as follows:

```
      SEGMENT [ = label ]
      [ALLOW {number | (number[,number]...)}]
      [DEFAULT number]
      CAS
              ( CA [= label]

                [ALLOW {number | (number[,number]...)}]
                [DEFAULT number]
                CUINTS = ( label [[number]] [,label [[number]]]... )
              )...

      [ALLOW {number | (number[,number]...)}]
      [DEFAULT number]
      DAS
              ( DA [= label]

                [ALLOW {number | (number[,number]...)}]
                [DEFAULT number]
                DEVINTS = ( label (,label)... )
              )...
```

The ALLOW clause lists the possible numbers of sub-objects that can be created. When an ALLOW clause is present, the maximum allowed number must match the number of sub-object definitions that follow.

The DEFAULT clause specifies the number of sub-objects that are created initially. If there is no default clause, then the first allowed value is used. If there is no ALLOW clause then the default is the number of sub-object definitions that follow.

The allowed values for subsystem attributes are:

Table 41. Controller PDF Attributes

Attribute	Allowed Values	Description
ABBREVID	String of up to 8 characters	An optional identifier that can be used to select a particular physical description during IODF load or re-synchronization.
UNITTYPE	String of up to 13 characters	A valid control unit type (required) and model (optional). (e.g. "3990-3")
COMGROUP	TRUE or FALSE	If TRUE, can be grouped via communicator cables. Default: "FALSE".
CANGROUPADAPTERS	TRUE or FALSE	If TRUE will respond to the HCM view preference that causes channel adapters and device adapters to be grouped together by adapter number. The default value is FALSE.
OWNSALLDEVICES	TRUE or FALSE	If TRUE controllers of this PDF are the sole owners of their devices. Used during IODF load and resynchronization to match controller segments. The default value is FALSE.

The standard IBM PDF for 3990-3 controllers is as follows:

```
SubSystem = "IBM 3990 Model 3 DASD Controller"
UnitType = "3990-3"
Class = "DASD"
OwnsAllDevices = True
Controller
Allow 2 Segments
Segment = "SC0"
Allow (2,4,6,8) Default 4 CuInts=(A,B,C,D,E,F,G,H)
Allow 2 DevInts=(SP0,SP1)
Segment = "SC1"
Allow (2,4,6,8) Default 4 CuInts=(A,B,C,D,E,F,G,H)
Allow 2 DevInts=(SP2,SP3)
Controller
Allow 2 Segments
Segment = "SC2"
Allow (2,4,6,8) Default 4 CuInts=(A,B,C,D,E,F,G,H)
Allow 2 DevInts=(SP0,SP1)
Segment = "SC3"
Allow (2,4,6,8) Default 4 CuInts=(A,B,C,D,E,F,G,H)
Allow 2 DevInts=(SP2,SP3)
```

Every controller has at least one device adapter, containing at least one device interface to connect to a string. Depending on the number of device adapters and device interfaces which you decide to include in the view (often this number equals the number of device adapters and device interfaces defined in the PDF file), either the adapter labels or the interface labels are shown in the diagram. Due to space limitations, only one kind of label is displayed. The decision which label is shown is made as follows:

Case 1: If at least one device adapter of the controller is labelled in the controller's current physical description, *and* if there are two or more device adapters defined

| where every device adapter has at most one device interface, the adapter labels are
| shown. Else, if two or more device interfaces are defined per device adapter, then
| the device labels are shown.

| *Case 2:* If no device adapter is labelled in the controller's current physical
| description, *and* if you have multiple device adapters, each containing multiple
| device interfaces, then no label is shown. Else, if you have at least two device
| adapters where each adapter has exactly one device interface, then the device
| interface is shown.

| *Case 3:* If the controller's PDF describes a controller which has one device adapter
| containing one device interface, the control unit number is shown instead of any
| label, regardless whether the device adapter or the device interface is labelled or
| not.

| The *controller's current physical description* is either the chosen physical
| description contained in a PDF file, or it is the physical description that you
| manually created for the controller by changing the number of device adapters and
| interfaces in the **Arrange Controller** dialog. If device labels are selected to be
| shown in the configuration diagram, HCM can only show a reasonable label if one
| is provided in the physical description file.

| Depending on what kind of label(s) you expect to see in the diagram at the bottom
| of your controller, you will have to set up the device adapter and device interface
| labels for your controller in the PDF files appropriately.

String PDF syntax

A string PDF is an ASCII text file that contains one or more string physical
descriptions. A string physical description has the following syntax.

```
STRING = label
[ABBREVID = id]
UNITTYPE = unit
[COLUMNS = number]
[TWOHEADED = {TRUE | FALSE}]

  [  (DEFAULT | EXACTLY) number UNITS   ]

  (   UNIT [= label]  (
DEVICES = number |...
DEVICERANGE = (position,count) [,(position,count)]...)
  )...

[ALLOW {number | (number[,number]...)}]
[DEFAULT number]
HOSINTS = ( label (,label)... )
```

Explanation of PDF changes

DEFAULT

> The DEFAULT clause for the number of units indicates the number of units
> the string is created with. If the EXACTLY clause is used, this number does
> not change during the string is installed. The number of the unit
> definitions must be equal or larger as the number specified in the clause -
> if EXACTLY is used, no unit definitions are allowed.

DEVICES DEVICERANGE

> The DEVICES and DEVICERANGE clauses in the unit definitions specify either a
> count of devices or ranges of devices. Each range of devices owns a
> position and count. If the first unit definition uses the DEVICES keyword, all

subsequent unit definition must use the `DEVICES` keyword. In the same way, if the first unit uses `DEVICERANGE`, all subsequent units must use `DEVICERANGE`. These definitions control the assignment of I/O device numbers to units when the string is created or when devices are added to the string. If the `DEVICES` keyword is used, new devices will be assigned to units in the string sequentially - each unit owns the specified number of devices. If `DEVICERANGE` is used, the position of the device in a sequence of new device numbers being added to the string will be compared with the `DEVICERANGE` definition for the units to determine which unit the device should be added to.

The allowed values for string attributes are:

Table 42. String PDF Attributes

Attribute	Allowed Values	Description
ABBREVID	String of up to 8 characters	An optional identifier that can be used to select a particular physical description during IODF load or re-synchronization.
UNITTYPE	String of up to 13 characters	A valid device type (required), e.g. "3390"
COLUMNS	Positive integer	A number indicating the number of units in each row (default is 1).
TWOHEADED	TRUE or FALSE	If TRUE, the string is drawn with two strings of units and two sets of interfaces. The default value is FALSE.

The IBM PDF for 3390 strings is as follows:

```
String = "IBM 3390 all models"
UnitType = "3390"
Default 3 Units
    Unit    Devices=8
    Unit    Devices=12
    Unit    Devices=12

Allow (4) Default 4 hosInts=(A,B,C,D)
```

The IBM PDF for IBM RAMAC Array DASD is as follows:

```
String = "IBM RAMAC Array DASD"
AbbrevId = "RAMAC"
UnitType = "3390"
Columns = 2
Exactly 16 Units
    Unit  DeviceRange=(30,2)
    Unit  DeviceRange=(28,2)
    Unit  DeviceRange=(26,2)
    Unit  DeviceRange=(24,2)
    Unit  DeviceRange=(22,2)
    Unit  DeviceRange=(20,2)
    Unit  DeviceRange=(18,2)
    Unit  DeviceRange=(16,2)
    Unit  DeviceRange=(0,2)
    Unit  DeviceRange=(2,2)
    Unit  DeviceRange=(4,2)
    Unit  DeviceRange=(6,2)
    Unit  DeviceRange=(8,2)
    Unit  DeviceRange=(10,2)
    Unit  DeviceRange=(12,2)
    Unit  DeviceRange=(14,2)

  Allow 4 hosInts=(A0,A1,A2,A3)
```

This indicates that a RAMAC string contains 16 units arranged in two columns in
the diagram, and that the first two devices (0 and 1) are assigned to the ninth unit
(fifth row and first column), and so on.

Chapter 18. Problem determination

The information in this chapter is intended to help you diagnose problems that may arise from using HCM. It explains:

- How to identify problems
- What diagnostic information and tools you can use
- How to search problem reporting data bases

Identifying and handling problems

The basic strategy for handling problems is as follows:

- Identify problem
- Diagnose problem
- Collect problem data, if applicable
- Analyze problem data
- Develop search argument
- Report problem to IBM
- Identify a problem source

More information on how to proceed in handling a problem can be found in

- *z/OS MVS Diagnosis: Procedures*
- *z/OS MVS Diagnosis: Reference*
- *z/OS MVS Diagnosis: Tools and Service Aids*
- *z/VM Diagnosis Guide*
- *z/VM System Messages and Codes*

Errors may occur while:

- Installing HCM at the workstation.
- Establishing a communication between the workstation and HCD at the host.
- Communicating with HCD at the host.
- Processing an HCM action
- Drawing the configuration diagram

The following topics are discussed:

- "Application error" on page 382
- "Fatal error at workstation" on page 382
- "Abnormal termination of HCD host" on page 386
- "Communication problems using APPC" on page 382
- "Communication problems using TCP/IP under z/OS" on page 383
- "Communication problems using APPC" on page 382 and "Communication problems using TCP/IP under z/OS" on page 383
- "Message problems" on page 384
- "Incorrect results" on page 385
- "Incorrect configuration diagram" on page 385
- "Problems printing reports/configuration diagrams" on page 385

- "Incorrect help information" on page 388
- "Problems with HCM User's Guide" on page 387
- "Installation problems" on page 388

Messages are the primary means by which HCM informs you that an error occurred. The message is either issued by HCM or originated from the HCD host. In some cases, an HCD message identifies errors reported by other products.

Product identifiers

- Module prefix: *EEQ*
- Component ID: *5797-11900*

For more information on identifiers, refer to the *z/OS Program Directory* or to the *HCD and HCM for z/VM Program Directory*.

Application error

An application error is a severe error that might occur while working with HCM. It causes the operating system to display a process exception with various information where the error occurred. From the information displayed, record the following:

- Type of exception (for example, general protection fault)
- Failing module
- Location within failing module
- Dr. Watson log file (if available)

and report the problem to the IBM Support Center.

Fatal error at workstation

If HCM detects a "logic error", it abends with message EEQ0000T and an error code. This error code is used to determine the reason of the error.

Provide the following information to the IBM Support Center.

1. Task to be accomplished
2. Name of the dialog box in error (title)
3. Sequence of events that led you to the failing dialog
4. Complete error code and the HCMERR.LOG PWS file (see "The HCMERR.LOG PWS file" on page 390)

Communication problems using APPC

There are a variety of reasons why HCM might fail to setup an APPC session with the host.

Possible causes are:

- Improper definitions in the node definition file (.NDF) on the workstation
- Incorrect user id
- Incorrect password or password expired
- Host not available

You may want to follow the hints listed below:

1. Examine the message explanation

2. Make sure that
   - VTAM is started on the host
   - APPC is started on the host
   - The APPC scheduler is started
   - TP-profile defined for HCM
3. Examine the host APPC log data set "userid.Annnnnnn.Dddddd.Tttttt.JOBLOG" (nnnnnnn is a decimal number). This error log will be created if the APPC scheduler detected an error condition, for example the job defined in the TP-profile had a JCL error. The name of the APPC log data set is defined in the TP-profile.

   For an example of a TP-profile, see Figure 5 on page 15.

Possible causes for the loss of communication while exchanging data with the HCD host are:
- Host termination
- HCD abend
- Network problem

See also the host APPC error log data set "userid.Annnnnnn.Dddddd.Tttttt.JOBLOG" (nnnnnnn is a decimal number) if you are running APPC. The data set contains information about the job that runs on the host.

The PCOMM trace or VTAM trace can also be useful for error determination.

Note: A host running under z/VM does not support APPC for the communication between HCM and HCD.

Communication problems using TCP/IP under z/OS

There are a variety of reasons why HCM might fail to setup a TCP/IP connection with the host running under z/OS.

Possible causes can be:
- TCP/IP is not up and running.
- The HCM user ID is not authenticated for use by UNIX System Services or does not have a home directory.
- A DD statement for SYSTCPD is not available.
- The HCM agent job is not started as a lack of initiators.
- There are problem with the names server.

The following suggestions may help to pinpoint the cause of the problem:
- Issue a **ping** command to check whether you have a TCP/IP connection from your workstation to the host.
- Observe the console output for CBDGnnnI messages when you start the HCM dispatcher.
- Try to login with HCM using a wrong user ID or password: You should get an EEQX401I or EEQX402I message. In addition, you should see the CBDG985I message on the console.
- When using a correct user ID and password, check on the console, whether an HCM Agent job has been started, check for CBDGnnnI messages. If the HCM

agent job has terminated, check its job output for possible reasons, check whether there are any EDCnnnI messages.

For additional information, please refer also to the *TCP/IP Support* documentation, which can be found on the HCM home page:

http://www.ibm.com/servers/eserver/zseries/zos/hcm
– – – > **Product Updates**
– – – > **HCM Documentation**

Communication problems using TCP/IP under z/VM

There are a variety of reasons why HCM might fail to setup a TCP/IP connection with the host running under z/VM.

Possible causes can be:
- TCP/IP is not up and running on the host and/or on the workstation.
- The z/VM HCD TCP/IP dispatcher for HCM is not up and running on the host.
- The external security manager (ESM) is not properly configured on the host.

The following suggestions may help to pinpoint the cause of the problem:
- Issue a **ping** command to check whether your host is reachable from your workstation.
- Check the output from the z/VM HCD TCP/IP dispatcher for HCM on the host.
- Try to login with HCM and check for messages from HCM on the workstation and from the z/VM HCD TCP/IP dispatcher for HCM on the host.
- Use the **TCP REPORT ON** command in the z/VM HCD TCP/IP dispatcher for HCM on the host to get information about TCP/IP traffic (refer to *Reusable Server Kernel Programmer's Guide and Reference* for details).
- Use the **TCP LIST** and **TCP QUERY subtaskid** commands in the z/VM HCD TCP/IP dispatcher for HCM on the host to get information about TCP/IP sessions (refer to *Reusable Server Kernel Programmer's Guide and Reference* for details).

Message problems

HCM issues messages that start with the product identifier *EEQ*. Typically, these messages warn you, or report an error condition. The messages are described in the appendix of this book.

Report any difficulties you have with an HCM message, for example:
- A message is wrong, missing or incorrectly formatted
- A message does not adequately explain the problem
- It is not clear why the message was issued

Provide the following information when reporting a message problem to the IBM Support Center:
1. Message identifier
2. Message text
3. Sequence of events leading to the message
4. Description of the problem

Incorrect results

You should report a problem when:

- HCM output is missing or contains unexpected information
- HCM fails to respond to the requested action, or incorrectly responds to the requested action without any message.
- The information/data displayed in a window is wrong or unclear

The following information is required when reporting the problem to the IBM Support Center:

1. Task to be accomplished
2. Sequence of windows (dialog boxes) used
3. Name (title) of the window in error
4. Last message displayed on the screen

The following is an attempt to give you some guidance for describing the problem:

- What operating system version are you running on (Windows 95 or higher)?
- What were the symptoms, if any?
- What are the other active programs?
- Is the problem recurring?
- What was the last sequence of events that led to the problem?
- Did these events run successfully in the past?
- If the problem is associated with a specific action, note the variables and parameters.

Incorrect configuration diagram

If you encounter that the configuration diagram is incorrectly drawn, report the problem to the IBM Support Center and provide the following information:

- Print Screen of the failing diagram or
- Bitmap of the configuration diagram
- Sequence of events that led to the failing diagram.

Problems printing reports/configuration diagrams

There are various things that can go wrong when printing a configuration diagram or context report, for example

- Wrong scaling
- Clipping
- Wrong line and fill patterns
- Wrong page breaks

The most common causes of printing problems is an incompatibility between the printer driver and the printer, or a defect in the printer driver. Make sure that the printer driver is the correct one for use with the printer you are using and that it is the latest version of the printer driver.

Problems with Tivoli Inventory support

Tivoli Inventory gathers information about HCM data by scanning MIF files. A conversion tool creates these files by converting HCM exported files to MIF files.

If an error occurs during the scan process, this information can be used to determine which component caused the problem.

If an error occurs during conversion, the MIF files EEQMIFxx.MIF in the target directory will be renamed to EEQMIFxx.MI~. The new extension prevents Tivoli Inventory from reading inconsistent MIF files. Tivoli Inventory will stop the processing of the distributed profile and report that some MIF files cannot be read. Take a look into the log file of the conversion tool to check the messages.

Abnormal termination of HCD host

If HCD abends on the host, it makes an attempt to recover and automatically start up again. HCM displays one or more messages describing the cause of the host abend, followed by any initialization message(s).

The abend may be caused
- By detecting a logical error (most likely indicated by means of message CBDA099I)
- Broken code

You can obtain more detailed information about the HCD abend by activating the HCD trace in the *Options* menu before re-attempting the task/action that caused the host abend. The HCD trace contains diagnostic information about the abend:
- System abend code + reason code
- HCD module stack
- HCD main entry point address
- PWS and registers
- HOM history table

The following information is required when reporting an HCD host abend:
1. Sequence of events leading to the abend, such as the HCM action being performed, dialog flow, etc..
2. HCD host trace (if available)
3. HCM message, if any

HCD host problems

You may be confronted with various failure causes on the host:
- Incorrect or incomplete data returned to HCD.
- Host failed to respond
- Missing or incorrect output
- Incorrect validation of the configuration change request
- Wrong HCD message

The best vehicle to determine whether or not the HCD host is working correctly, is the HCD trace. Activate the HCD trace prior to performing the action that caused the host problem. Refer to "Using the HCD trace facility" on page 388 for more details on how to use the HCD trace function.

Out of memory condition
If you receive one of the following HCD messages:
- CBDA056I There is not enough virtual storage available,
- CBDA474I Insufficient space to read IODF *dsname* into storage

you should consider providing more virtual storage for the HCD server on the host or enable HCD to load the IODF into a data space.

For z/OS, to increase the virtual storage adapt the REGION parameter for the HCD server as follows:

- If you use TCP/IP as communication protocol, you can either use an increased hardcoded value for the REGION size in the Skeleton (see "Skeleton used to start the HCM agent" on page 9), or edit the EEQHCM.INI file on the workstation and specify an appropriate value for the AREGION in the section [CBDQAGNT] (see "HCM client: configurable login parameters" on page 429).
- If you use APPC as communication protocol, you can increase the REGION size in the APPC Transaction Profile (see "Step 1: Define TP profile" on page 14).

For z/VM:
You can use DEFINE STORAGE commands to increase the available storage.

If you load the IODF into a data space, you will get storage relief in your address space. To enable HCD to load the IODF into a data space, you should specify IODF_DATA_SPACE=YES in your HCD profile on the host (refer to the *z/OS HCD User's Guide* or to the description of z/VM HCD in *z/VM: I/O Configuration*).

Problems when using I/O Operations

If you get unexpected results when working with HCM I/O Operations functions, and some requests to I/O Operations appear to have returned misleading or incomplete data, you can use the command interface to issue a DISPLAY RESULTS command to display the result of the most recently processed command. You can also check on the console, which I/O Operations commands have been issued by HCM.

Problems with HCM User's Guide

Report any incorrect or missing information in the HCM User's Guide, for example
- Essential information for HCM that is missing
- Any description of HCM processing that does not match the reality.
- Information is so vague that it prevents the effective use of HCM
- Conflicting information

If there is an error, ambiguity, or omission in the HCM User's Guide, determine if the situation is serious enough to affect the usage of HCM. For example, errors such as incorrect page reference or an ambiguity in a general description will not affect the usage of HCM. However, a wrong sequence in a series of steps that you have to go through can affect the HCM usage.

If the problem does not affect the usage of HCM, fill out the Reader's Comment provided at the back of the publication or write to the address shown in the edition notice. Comments help IBM improve future revisions of the publication.

If the problem has affected or can affect HCM usage, report the problem to the IBM Support Center, providing the following information:
1. Form number of publication
2. Information about the incorrect description (chapter title, page number)

Incorrect help information

If you encounter problems with content, wording or mismatches, report the following to the IBM Support Center.

- Name of the help window
- Name of the last displayed dialog box
- Short description of the action being processed

Problems with UIM-provided help

UIM provided help is not stored in the HCM help file (EEQHCM.HLP). This type of help will be made available to you by obtaining the help text from the host and converting it into a displayable format.

In case of an error, provide the following information to the IBM Support Center:

- Name of parameter/feature for which help was requested
- Type/model of the device for which the definition is made

Installation problems

If you have problems after starting the installation program, first check to see whether your files have been corrupted. It may be necessary to download them again in binary mode and retry the installation. If this doesn't help, obtain the current host PTF new and install it. Download the files again and retry the installation. If this still doesn't help, contact IBM.

Using the HCD trace facility

HCD provides a trace facility that records information that can be used to locate internal HCD problems. The HCD trace records all data received by the host, passed back to the workstation, as well as the flow between the various HCD modules within the host.

You can enable the HCD trace facility from HCM. After you have logged on, you can turn the HCD trace facility on and off using the *Activate HCD Trace* menu item under *Options*. Thus you can capture trace information for only the operations that are of interest.

To turn on the HCD trace facility before logging on, select the HCD Trace checkbox in the Login dialog. You should only turn on tracing before logging on if you are interested in tracing the logon process itself. The logon process generates a large amount of trace information that you would otherwise not need. To use the HCD trace facility under z/OS, you must pre-allocate a trace data set with a DD name of **HCDTRACE**.

Note: When the trace facility is turned on, all the trace categories will be traced until the HCD profile is read. At this point, trace parameters set in the HCD profile will influence the trace behavior. If the trace facility is turned on later, only the trace parameters set in the HCD profile will be used. See the section on Diagnostic Information and Tools in the applicable *z/OS HCD User's Guide* or in the *z/VM: I/O Configuration* for further information.

Using the z/VM HCD TCP/IP Dispatcher for HCM trace facility

To use the z/VM TCP/IP Dispatcher trace facility, start it with the command CBDIODSP TRACE START. Trace filename, filetype, and location (minidisk or SFS directory) are specified by defining the names CBDTRANT (name and type) and CBDTRAMD (minidisk or directory) in the CBDCONF NAMES file on your A-disk. For more information on the z/VM HCD TCP/IP dispatcher for HCM trace see the following section "z/VM HCD TCP/IP Dispatcher for HCM commands."

z/VM HCD TCP/IP Dispatcher for HCM commands

The z/VM HCD TCP/IP dispatcher for HCM recognizes the two command families TRACE and SECURITY which can be issued via RSK. The dispatcher accepts commands via the CONSOLE and SUBCOM line drivers.

- CBDIODSP TRACE START: Starts dispatcher trace. If trace is already started, nothing happens. Trace filename, filetype, and location (minidisk or SFS directory) are specified by defining the names CBDTRANT (name and type) and CBDTRAMD (minidisk or directory) in the CBDCONF NAMES file on your A-disk. If the trace file already exists when starting the trace, it is erased. The trace file is opened for writing and a starting timestamp is written into the trace. Trace entries will be written to the trace according to the trace category and detail settings in effect. CBDIODSP TRACE START does not affect trace category and detail settings. By default, the trace is started when starting the dispatcher.

- CBDIODSP TRACE STOP: Stops dispatcher trace. If trace is already stopped, nothing happens. The trace file is closed and a closing timestamp is written into the trace. Trace entries will not be written to the trace file any more until the trace is started again (but the current content of the trace file will be erased.) CBDIODSP TRACE STOP does not affect trace category and detail settings.

- CBDIODSP TRACE LIST: The dispatcher prints the trace category and detail settings which are currently in effect.

- CBDIODSP TRACE CATEGORY <category-name> (ON | OFF): Switches the category specified by <category-name> on or off. Valid categories are:
  - DISPATCHER
  - WORKER
  - RECEIVE
  - SEND
  - CRYPTO
  - DATAGRAM
  - COMMAND
  - INITIALIZATION
  - TERMINATION
  - RANDOM
  - HOM
  - UTILITY

- CBDIODSP TRACE DETAIL <detail-name> (ON | OFF): Switches the detail specified by <detail-name> on or off. Valid details are:
  - DEBUG
  - ERROR
  - ABEND
  - CONFIDENTIAL

- ENTRYEXIT
- MESSAGE
- CBDIODSP TRACE NONE: Switches off all category and detail settings supported by the dispatcher.
- CBDIODSP TRACE ALL: Switches on all category and detail settings supported by the dispatcher.
- CBDIODSP SECURITY CP: Enforces logon verification according to CP security (DIAG88 is used). This setting is default.
- CBDIODSP SECURITY ESM: Enforces logon verification according to ESM settings. ESM must be initialized.
- CBDIODSP SECURITY LIST: Prints the security model in effect for the dispatcher (CP or ESM).

The HCMERR.LOG PWS file

HCM incorporates another aid for problem determination. If a severe, unexpected, fatal error occurs, HCM saves its current execution environment in a text file called HCMERR.LOG.

This file is written by default to the same directory from which the HCM executable was launched. However, you can determine another than the default directory. This is useful if you have HCM installed in a LAN environment that has read access only. To keep the HCMERR.LOG on your local resource, you have to specify a directory in your EEQHCM.INI file. HCM uses the EEQHCM.INI file to store information and defaults across HCM sessions. It is located in the Windows system directory. Enter a line like the following into the EEQHCM.INI file in the [HCM] section:

```
ERRLOGDIR=C:\HCMLOG
```

The file is initially created by HCM when the first fatal error occurs. You can truncate or delete the file. If you do not delete the file, the data of the execution environment is appended for each fatal error.

If you want to report a problem to IBM, check whether HCM has written the file. If the file exists, supply the file to IBM.

Searching problem reporting databases and reporting problems

Search arguments are used to search problem reporting databases. If the problem being diagnosed was already reported and the symptoms entered into the database, the search will produce a match.

To perform a search, do the following:

1. Analyze the problem reporting data base and develop a search argument using the information provided in the boxes labeled *Search Argument*.
2. Complete the digits (such as ccc, nnn, hhh) according to the applicable conditions. For example, if the message EEQ0000T was received, the developed search argument for *message identifier* would be: MS/EEQ0000T. An example is shown below.

Search Argument	Description	Example
MS/ccccnns	Message identifier	MS/EEQ0000T

3. Use the search arguments to search *problem reporting data bases*. If the search finds that the problem has been reported before, request a fix from IBM.

   If the search is unsuccessful, report the problem to the IBM Support Center. Submit the information that is listed individually for the various problem areas.

For more detailed information on these steps, refer to *z/OS MVS Diagnosis: Procedures* or to the *z/VM Diagnosis Guide*.

Reporting problems to IBM

Before reporting a problem to IBM, search the problem-reporting data bases for the problem and possible solutions. If the problem cannot be resolved using the problem-solving database, then assemble the following information as appropriate before contacting IBM:

- Provide the data as stated in the appropriate section for the type of problem.
- Operating system type and version you are running HCM on.
- Version of HCD on the host
- Service Level of HCM
- The type of connection between the host and PWS including all software names and versions.
- The last message that HCM displayed in the status bar (bottom of window), particularly if HCM abends during IODF load/resync.
- The exact circumstances and actions that lead up to the error.
- Information about content and size of the used IODF and config files.

  In certain cases. IBM might ask you for the config file and associated IODF in order to re-create the problem.
- An HCD trace file of the problem. See "Using the HCD trace facility" on page 388 for more information.
- The HCMERR.LOG file, if it exists. Refer to "The HCMERR.LOG PWS file" on page 390 for details.

Be prepared to re-create the error and, if possible, to give IBM copies of your IODF and config file (.HCM).

Appendix A. HCM messages

Note: Messages beginning with CBD are issued by HCD. Refer to *z/OS and z/VM HCD Messages* for an explanation of these messages.

EEQ0000T **Fatal Error. Please contact IBM technical support and refer to this error code:** *error-code.* **Additional diagnostic information is available in the HCMERR.LOG file, if it exists.**

Explanation: HCM detected a logic error. The cause of the error is described in the *error-code*. The *error-code* contains the name of the module and the statement number where the logic error was detected.

System Action: HCM will terminate.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ0200E **The PWS config file has not been saved. You may retry after rectifying the problem (for example by switching to File Manager and freeing some disk space). If you Cancel the file will not be saved.**

Explanation: The PWS config file has not been saved.

System Action: Waits for user to try again or to cancel.

User Response: You may retry after rectifying the problem (for example by switching to File Manager and freeing some disk space).

EEQ0201E **Error opening PWS config file** *filename* **for read. Access denied. The file may be in use by another user.**

Explanation: The PWS config file could not be opened.

System Action: Aborts the file I/O operation.

User Response: Obtain access to the PWS config file and repeat the file I/O operation.

EEQ0202E **Operating system error opening PWS config file** *filename* **for read.** *OS-error-id: OS-error-message*

Explanation: Consult the documentation for your PWS operating system for the *OS-error-id* and *OS-error-message*.

System Action: Aborts the file I/O operation.

User Response: Correct the condition described by the OS error and repeat the file I/O operation.

EEQ0203E **Error opening PWS config file** *filename* **for update. Access denied. The file may be in use by another user or be write-protected.**

Explanation: Ensure that the config file can be opened for update.

System Action: Aborts the file I/O operation.

User Response: Obtain access to the PWS config file and repeat the file I/O operation.

EEQ0204E **Operating system error opening PWS config file** *filename* **for update.** *OS-error-id: OS-error-message*

Explanation: Consult the documentation for your PWS operating system for the *OS-error-id* and *OS-error-message*.

System Action: Aborts the file I/O operation.

User Response: Correct the condition described by the OS error and repeat the file I/O operation.

EEQ0205T **Re-open of PWS config file** *filename* **for read failed. Access denied. The file may be in use by another user. HCM will terminate.**

Explanation: Ensure that the config file can be opened for read.

System Action: HCM terminates.

User Response: Obtain access to the PWS config file and repeat the file I/O operation.

EEQ0206E **Operating system error re-opening PWS config file** *filename* **%d: %s HCM will Terminate.**

Explanation: Consult the documentation for your PWS operating system for the *OS-error-id* and *OS-error-message*.

System Action: Aborts the file I/O operation.

User Response: Correct the condition described by the OS error and repeat the file I/O operation.

EEQ0207E Error opening temporary PWS config file *filename*. **Perhaps the disk is write protected.**

Explanation: HCM cannot open a temporary PWS config file.

System Action: Aborts the file I/O operation.

User Response: Ensure that temporary files can be created and written. Consult the documentation for your PWS to determine where temporary files are created.

EEQ0208E Error trying to backup PWS config file *filename*. **The changes have been saved to temporary file** *filename*.**.**

Explanation: HCM cannot backup the PWS config file.

System Action: Aborts the file I/O operation.

User Response: Ensure that the PWS config file an be created and written.

EEQ0209E Error trying to backup PWS config file *filename*. **This file has been renamed to** *filename_2* **and the changes have been saved to temporary file** *filename_3*.

Explanation: HCM cannot create the PWS config file.

System Action: Aborts the file I/O operation.

User Response: Ensure that the PWS config file an be created and written.

EEQ0210E The changes to PWS config file *filename* **have been saved, however the temporary file** *filename* **could not be removed.**

Explanation: HCM cannot delete a temporary PWS config file.

System Action: Completes the file I/O operation, except for deleting the named temporary file.

User Response: Delete the named temporary file.

EEQ0211E Error opening PWS config file *filename*. **Perhaps the disk is write protected.**

Explanation: HCM cannot create the PWS config file. Occurs during a copy operation.

System Action: Aborts the file I/O operation.

User Response: Ensure that the PWS config file an be created and written.

EEQ0212E Operation not allowed in Standalone Mode.

Explanation: Connecting and disconnecting interfaces, as well as operations which require communication with the host, are not allowed when running HCM in Standalone Mode.

System Action: Aborts the attempted operation.

User Response: Open the PWS file again when HCM is not in Standalone Mode and attempt the operation again.

EEQ0213E Operation not allowed for production PWS files.

Explanation: Certain operations, such as editing, are forbidden when the currently open PWS file is a production file.

System Action: Aborts the attempted operation.

User Response: Convert the file to a work file using the File | Make Work... menu choice and perform the operation on the work file.

EEQ0214E This configuration is open in Read-Only mode. Updates are not allowed.

Explanation: When the PWS file was opened, it was opened with read-only access. This is either because you selected read-only access from a re-synchronization or upgrade confirmation dialog, or because the configuration contains advanced features saved by a later release of HCM, which this release would not be able to maintain if editing were permitted.

System Action: Aborts the attempted operation.

User Response: Determine the reason for the read-only access. Closing, then opening the PWS file again should cause a dialog to be displayed which will indicate the reason. If the reason was that you had chosen read-only access from a confirmation dialog, you may choose read-write access instead. If the reason was that the configuration contains advanced features, you have two options:
1. Remove the advanced features from the IODF and re-synchronize the PWS file to the IODF. It will then be editable.
2. Upgrade to a newer release of HCM.

EEQ0215E Necessary PWS file upgrading could not be completed.

Explanation: PWS files saved with an earlier version of HCM sometimes require upgrading (information conversion and supplementation) when opened with a later version. The above message means the upgrading could not be completed (often because host communication or actions failed, or write access to the PWS file could not be gained). This message usually occurs in conjunction with another message which

more specifically states the reason for the failure.

System Action: HCM aborts the attempt to open the PWS file and waits for next user menu selection.

User Response: Alleviate the cause of the failure and try opening the PWS file again.

EEQ0250E   **File** *filename* **line** *line-number*: **Import terminated.** *Variable text*.

Explanation: The import operation can not continue due to one or more severe errors.

System Action: HCM aborts the import operation and returns to the main diagram with an empty configuration (no file open).

User Response: Alleviate the cause of the failure and try importing the file again.

EEQ0251E   **File** *filename* **line** *line-number*: **Record not processed.** *Variable text*.

Explanation: The record data is ignored due to one or more errors.

System Action: HCM asks you if you want to keep the imported changes.

User Response: Respond with "no" to return to the main menu, respond with "yes" to proceed with the IODF update dialogs.

EEQ0252E   **File** *filename* **line** *line-number*: *field-description* **not processed.** *Variable text*.

Explanation: Aside from the field, the record data was used.

System Action: HCM asks you if you want to keep the imported changes.

User Response: Respond with "no" to return to the main menu, respond with "yes" to proceed with the IODF update dialogs.

EEQ0253I   **File** *filename* **line** *line-number*: *Variable text*.

Explanation: Indicates side effects that can occur while processing an import record, for example, the implicit disconnect that occurs when two otherwise connected interfaces are connected with each other.

System Action: Processing proceeds with the IODF Update dialog.

User Response: No action.

EEQ0260E   *Variable text*.

Explanation: The export operation can not continue due to one or more severe errors.

System Action: HCM aborts the export operation and returns to the main diagram.

User Response: Alleviate the cause of the failure and try exporting the file again.

EEQ0261E   *Variable text*.

Explanation: The record data is ignored due to one or more errors.

System Action: No action

User Response: No action

EEQ0262E   *Variable text*.

Explanation: Aside from the field, the record data was used.

System Action: No action

User Response: No action

EEQ0263I   *Variable text*.

Explanation: Indicates side effects that can occur while processing an export record.

System Action: Processing proceeds.

User Response: No action.

EEQ0400E   **Error opening controller physical description file:** *filename*

Explanation: HCM cannot open a PDF file.

System Action: Waits for user to either cancel or continue, in which case the PDF is ignored.

User Response: Ensure that HCM can open the PDF files in the controller PDF directories.

EEQ0401E   **Error opening string physical description file:** *filename*

Explanation: Ensure that HCM can open the PDF files in the string PDF directories.

System Action: Waits for user to either cancel or continue, in which case the PDF is ignored.

User Response: Ensure that HCM can open the PDF files in the controller PDF directories.

EEQ0402E **Physical description file** *filename* **cannot be loaded.** *specific syntax error and line number.* **The rest of this PDF will be ignored. Do you wish to continue?**

Explanation: A syntax error has been found in a controller PDF.

System Action: Waits for user to either cancel or continue, in which the PDF is ignored.

User Response: Correct the error in the PDF file. Consult the syntax grammar for controller PDFs under "Controller PDF syntax" on page 374.

EEQ0403E **Physical description file** *filename* **cannot be loaded...** *specific syntax error and line number.* **This may cause problems when creating strings. Do you wish to continue?**

Explanation: A syntax error has been found in a string PDF.

System Action: Waits for user to either cancel or continue, in which case the PDF is ignored.

User Response: Correct the error in the PDF file. Consult the syntax grammar for string PDFs under "String PDF syntax" on page 377.

EEQ0404E **Subsystem type** *typename* **has already been updated**

Explanation: A second definition for this subsystem type was found in a controller PDF.

System Action: Waits for user to either cancel or continue. If the user cancels then the file I/O operation is aborted. If the user continues, the second PDF is ignored.

User Response: Either cancel and fix the problem or continue.

EEQ0405E **Subsystem type** *typename* **has already been defined in a previous PDF.**

Explanation: A second definition for this string type was found in string PDF.

System Action: Waits for user to either cancel or continue. If the user cancels, then the file I/O operation is aborted. If the user continues, the second PDF is ignored.

User Response: Either cancel and fix the problem or continue.

EEQ0410W **Could not open file** *apd-file-name*.

Explanation: The APD file could not be opened.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that the APD file could not be opened, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may choose to continue the load or re-synchronization or return to the dialog for choosing an APD file. Possible reasons that the file could not be opened are:
* incorrect file or directory specification
* lack of read access to the device, directory, or file
* file sharing conflicts.

It is often possible to remedy such problems by switching tasks, using a utility that allows file system manipulation, and then returning to the HCM task to choose the same or another APD file.

EEQ0411W **Line** *line-number:* **Control unit** *cu-id* **does not exist in the IODF.**

Explanation: In the APD file, a control unit id was encountered which does not exist in the IODF that HCM is using in the current load or re-synchronization operation.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog and choose another APD file. You may wish to use task switching to edit and correct the APD file before choosing the same APD file again.

EEQ0412W **Line** *line-number:* **device** *dev-id* **does not exist in the IODF.**

Explanation: In the APD file, a device string id was encountered which does not exist in the IODF that HCM is using in the current load or re-synchronization operation.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog and choose another APD file. You may wish to use task

switching to edit and correct the APD file before choosing the same APD file again.

EEQ0413W   Line *line-number:* **Controller physical description** *descrip-name* **not found.**

Explanation: A physical description name for a controller was specified in the APD file, but that name was not found in the physical description files (PDFs) available to HCM.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog and choose another APD file. You may wish to use task switching to edit and correct the APD file before choosing the same APD file again.

EEQ0414W   Line *line-number:* **String physical description** *descrip-name* **not found.**

Explanation: A physical description name for a device string was specified in the APD file, but that name was not found in the physical description files (PDFs) available to HCM.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog and choose another APD file. You may wish to use task switching to edit and correct the APD file before choosing the same APD file again.

EEQ0415W   Line *line-number:* **Invalid syntax:** *text-from-APD-file***.**

Explanation: The APD file contained text which could not be parsed because it did not conform to APD file syntactic requirements.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog

and choose another APD file. You may wish to use task switching to edit and correct the APD file before choosing the same APD file again.

EEQ0416W   Line *line-number:* **Too many errors were encountered - the rest of this file will be ignored.**

Explanation: Too many errors were found in the APD file to continue to parse it, which may indicate that the file was not a proper APD file at all.

System Action: HCM offers you the choice of proceeding with the IODF load or re-synchronization despite the fact that there were problems with the APD file, or returning to the *Choose Physical Description Assignment File* dialog.

User Response: You may continue with the IODF load or re-synchronization despite the errors or discrepancies in the APD file, or you may go back to the *Choose Physical Description Assignment File* dialog and choose another APD file. You may wish to use task switching to edit and correct the APD file before choosing the same APD file again.

EEQ0600E   Logon (HCD session establishment) error has occurred.

Explanation: This message is preceded by another message which tells why the logon attempt failed.

System Action: HCM terminates.

User Response: Remedy the situation that caused the logon attempt to fail (e.g. wait for the host system to become available).

EEQ0601E   Illegally formatted prompt information returned by HCD host.

Explanation: HCM has sent a *prompt* request to the HCD host, but the returned data are either incomplete, unrecognized or wrong. One possible reason is an HCD abend on the host or the loss of the communication link between HCM and HCD.

System Action: HCM terminates

User Response: See Chapter 18, "Problem determination," on page 381 for diagnostic instructions.

EEQ0602E   HCD on host has responded with wrong language (not English).

Explanation: HCD is not correctly configured.

Explanation: An attempt has been made to establish a session with HCD on the host, but the English language part is not installed on the host.

System Action: HCM terminates

User Response: Make sure that the English FMID is installed, otherwise see Chapter 18, "Problem

determination," on page 381 for diagnostic instructions.

EEQ0603E **HCD on host is running under wrong operating system.**

Explanation: An attempt has been made to establish a session with HCD on the host, but HCD does not run under MVS/ESA. This, however, is not possible.

System Action: HCM terminates

User Response: See Chapter 18, "Problem determination," on page 381 for diagnostic instructions.

EEQ0604E **An incompatible level of HCD is running on the host.**

Explanation: Ask your system programmer to upgrade HCD. For more information see the EEQREAD.ME file of HCM. Internal description code: Cw-x Ry-z.

System Action: HCM terminates.

User Response: Do not retry the action unless and until a newer release of HCD is installed on the host. The internal description code allows IBM Support to determine the required release of HCD.

EEQ0605W **The release of the HCD interface available on the host is not up to date with this release of HCM.**

Explanation: You may proceed, but some functionality may not be available.

System Action: None.

User Response: Proceed but expect some requested actions to fail with error.

EEQ0606E *feature-description* **not supported by the HCD interface for the release of HCD in use.**

Explanation: Ask your system programmer to upgrade HCD if applicable. For more information see the EEQREAD.ME file of HCM. Internal description code: Cw-x Ry-z.

System Action: HCM cancels the current action.

User Response: Do not retry the action unless and until a newer release or service of HCD is installed on the host. The internal description code allows IBM Support to determine the required release of HCD.

EEQ0607E **Attempt to load library EEQRHOMW.DLL failed. (You may still choose to run in Standalone Mode).**

Explanation: An attempt has been made to establish a host connection, but the library file EEQRHOMW.DLL cannot be loaded.

System Action: HCM displays the logon panel to allow another logon attempt.

User Response: Remedy the situation that caused the logon attempt to fail or choose to press the Standalone button if logging in is not possible or not desired.

EEQ0608E **GetProcAddress failed on EEQRHOMW.DLL. Your version of EEQRHOMW.DLL may be outdated or corrupt. (Replace EEQRHOMW.DLL or remove it and run in Standalone Mode).**

Explanation: An attempt has been made to establish a host connection, using an outdated or corrupt version of the library file EEQRHOMW.DLL

System Action: HCM displays the logon panel to allow another logon attempt.

User Response: Remedy the situation that caused the logon attempt to fail or choose to press the Standalone button if logging in is not possible or not desired.

EEQ0609W **The version of HCD on the host does not fully support switches with multiple internal control units (present in this IODF). Some switch related operations may be refused by HCD.**

Explanation: See message text.

System Action: None.

User Response: Do not retry the action unless and until a newer release of HCD is installed on the host.

EEQ0610E **The version of HCD on the host does not fully support the following CHPID types in this configuration file:** *filename***. HCM will refuse to open this configuration file when running online, however, you may open it in Standalone mode.**

Explanation: HCD does not fully support some CHPID types which are defined in the configuration. Ask you system programmer to update HCD.

System Action: HCM cancels the current action when running online, and HCM continues if running in standalone mode.

User Response: Do not retry the action when running online unless and until a newer release of HCD is installed on your host.

EEQ0611E **The version of HCM is incompatible with the level of the HCD which is running on the host.**

Explanation: Ask your system programmer to upgrade HCM as HCM does not support the version of HCD on the host. Internal description code: Cw-x Ry-z.

System Action: HCM cancels the current action.

User Response: Do not retry the action unless and until a newer release or service level of HCM is installed on your PWS. The internal description code allows IBM Support to determine the required release of HCM.

EEQ0991T   Invalid HOM storage descriptor. The program will terminate.

Explanation: While retrieving data from HCD, HCM received an unrecognized record.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ0992T   No data conversion defined for this storage descriptor. The program will terminate.

Explanation: The HCM ASCII/EBCDIC conversion routine received a record, but does not know how to convert the record. Most likely, a wrong record was received from HCD or passed for transmission to the host.

System Action: HCM terminates

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ0993T   Corrupted HOM interface record. The program will terminate.

Explanation: A corrupted record was received from HCD.

System Action: HCM terminates

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ0994T   A severe error has occurred on the host.

Explanation: HCD is not active on host. Program will terminate.

System Action: HCM terminates.

User Response: Run HCM again, this time selecting the checkbox for the TRACE option in the login dialog. Supply the resulting HCD.TRACE data set on the host to the IBM service center. Further clues to the cause of the severe error may be gained by running HCD manually on the host. If HCD experiences a similar severe error when run manually, it may be due to the same cause. There may be relevant messages left in the HCD.MSGLOG data set after a manual HCD session has terminated.

EEQ1000E   RHOM_E_NO_CONVERSATION - internal error. RHOM interface violation: Bad conversation handle.

Explanation: HCM passed an invalid conversation handle to RHOM during an HCD access call.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ1001E   RHOM_E_ALLOC - Insufficient memory (component RHOM)

Explanation: RHOM was unable to allocate memory (usually during host logon).

System Action: HCM terminates.

User Response: Try closing other Windows applications to free memory.

EEQ1002E   RHOM_E_NO_IBUF - Internal error. RHOM interface violation: Input buffer must not be NULL.

Explanation: The RHOM interface was violated by HCM by not passing a valid input buffer for HCD access.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ1003E   RHOM_E_NO_OBUF - Internal error. RHOM interface violation: Output buffer must not be NULL

Explanation: The RHOM interface was violated by HCM by not passing a valid output buffer for HCD access.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ1100E   RFHOM_E_BAD_REQUEST_CODE - Internal error. RFHOM interface violation: Unknown request.

Explanation: HCM/HCD interface routine received an invalid request code from the workstation. The reason is either a garbled network transmission or an HCM/HCD interface routine interface violation by component RHOM.

System Action: HCM terminates.

User Response: Report this problem to IBM. See

"Reporting problems to IBM" on page 391 on how to do that.

EEQ1101E **RFHOM_E_OPENLOG - Error opening HCD server host logfile**

Explanation: HCM/HCD interface routine received a request with the host logfile flag set. The attempt to open the logfile failed. The default logfile data set name on the host is HLQ.CBDQRHOM.LOG (where HLQ is the user's TSO session qualifier).

System Action: HCM terminates.

User Response: Verify that sufficient disk space on the host is available.

EEQ1102E **RFHOM_E_CLOSELOG - Error closing HCD server host logfile**

Explanation: HCM/HCD interface routine was unable to close the host logfile data set. The default logfile data set name on the host is HLQ.CBDQRHOM.LOG (where HLQ is the user's TSO session qualifier).

System Action: HCM terminates.

User Response: Verify that sufficient disk space on the host is available.

EEQ1103E **RFHOM_E_LOADHOM - Error loading HOM. Please verify that HCD host access is correctly configured. Check STEPLIB in APPC TP profile.**

Explanation: HCM/HCD interface routine could not load the HCD HOM interface component (CBDMGHOM).

System Action: HCM terminates.

User Response: Verify that HCD is correctly installed on the MVS host. Verify that the STEPLIB specification of your APPC TP profile includes the HCD load libraries.

EEQ1104E **RFHOM_E_DUMPHRB - Internal error (component RFHOM). Error during DumpHRB**

Explanation: HCM/HCD interface routine encountered an error in the DumpHRB routine.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ1105E **RFHOM_E_DUMPIOBUF - Internal error (component RFHOM). Error during DumpIOBuf**

Explanation: HCM/HCD interface routine encountered an error in the DumpIOBuf routine.

System Action: HCM terminates.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391 on how to do that.

EEQ5000I **Conversion routine started.**

Explanation: This message is shown when the routine started working. The input parameters are shown as they were stated.

EEQ5001I **Conversion routine ended.**

Explanation: This message is shown when the routine ended working and no error occurred that forced an abnormal or error end.

EEQ5003E **Files found: [Path+filename]**

Explanation: Shows a list of identified filenames including path which will be processed.

EEQ5004E **No file found in directory [Directory]**

Explanation: No files were found in the directory.

System Action: The conversion tool stops. No conversion took place.

User Response: Make sure that you did not specify an empty source directory.

EEQ5005E **No file found matching: '[File]'.**

Explanation: No file was found matching the name [File].

System Action: The conversion tool stops. No conversion took place.

User Response: Make sure you specified an existing source file.

EEQ5006I **Started processing of file: '[Path+filename]'**

Explanation: The mentioned file is being processed.

EEQ5007I **Ended processing of file: '[Path+filename]. All [X] data-records correctly processed.**

Explanation: The mentioned file has been processed. There were all [X] data records correctly identified.

EEQ5008E   Aborted processing file: '[Path+filename]'.

Explanation: The processing of the mentioned file was aborted due to the occurrence of an error.

System Action: Processing of the specified file stops. Conversion process continues.

User Response: Make sure, you specified only HCM exported files.

EEQ5010E   Unable to open file: '[Path+filename]'.

Explanation: There was an error while opening a file.

System Action: Conversion process stops.

User Response: Make sure, the user running the conversion tool has sufficient access authority to perform the action.

EEQ5011E   Unable to access directory: '[Directory]'.

Explanation: There was an error while trying to access a directory.

System Action: Conversion process stops.

User Response: Make sure, the user running the conversion tool has sufficient access authority to perform the action.

EEQ5012E   Unable to create file: '[Path+filename]'.

Explanation: There was an error while creating a file.

System Action: Conversion process stops.

User Response: Make sure, the user running the conversion tool has sufficient access authority to perform the action.

EEQ5013E   Unable to create directory: '[Directory]'.

Explanation: There was an error while trying to create a directory.

System Action: Conversion process stops.

User Response: Make sure, the user running the conversion tool has sufficient access authority to perform the action.

EEQ5014E   Unable to write to file: '[Path+filename]'.

Explanation: There was an error while trying to write to a file.

System Action: Conversion process stops.

User Response: Make sure, the user running the conversion tool has sufficient access authority to perform the action.

EEQ5015E   Invalid syntax.

Explanation: The conversion tool was invoked with a syntax error.

System Action: Conversion process stops. No conversion took place.

User Response: Make sure the conversion tool was syntactical correctly invoked.

EEQ5016E   No such logical drive: '[Drive]'.

Explanation: The source or target parameter referred to a nonexistent logical drive.

System Action: Conversion process stops. No conversion took place.

User Response: Make sure the specified source or target directory contains a valid logical drive letter.

EEQ5017E   Could not find [Directory].

Explanation: Unable to find a directory.

System Action: Conversion process stops. No conversion took place.

User Response: Make sure the specified source directory exists.

EEQ5018E   Is not a directory: '[String]'

Explanation: The specified target is not a directory.

System Action: Conversion process stops. No conversion took place.

User Response: Make sure a directory was specified when using the -t option.

EEQ5019E   Unable to allocate required memory.

Explanation: The process could not get sufficient memory from the run time system.

System Action: Dependent on the severity of the error the conversion process may either stop or continue.

User Response: Make sure that enough virtual memory is available from the runtime system.

EEQ5021E   Internal error.

Explanation: An error occurred due to the inner structure of the conversion tool.

System Action: Conversion process stops.

User Response: Report this problem to IBM. See "Reporting problems to IBM" on page 391.

EEQ5022I   MIF files may be corrupted.

Explanation: Due to an error during the conversion process, the integrity of the produced MIF files is not granted.

User Response: Check log file for error or warning messages, remove their cause, remove produced MIF files and run the Inventory scan and/or conversion process again.

EEQ5023E   Error while reading from file :'[File]'.

Explanation: An error occurred while reading from file [File]. The processing of this file will be aborted.

System Action: Conversion process stops.

User Response: Check the specified file and associated file system for integrity.

EEQ5024E   Unable to get required system information.

Explanation: The conversion tool could not get required information from the runtime system.

System Action: Conversion process stops.

User Response: Check setting and integrity of the operating system.

EEQ5025E   Unable to get file handle.

Explanation: The conversion tool could not get a file handle to search for specific files.

System Action: Conversion process stops.

User Response: Check uses file system and access authorities.

EEQ5026W   At least one error or warning message occurred during conversion.

Explanation: At least one error or warning message came up during the conversion process. For details see screen output or log file (if such one is produced).

User Response: Examine log file for error and/or warning messages and remove their cause. If necessary, remove already produced MIF files and run the Inventory scan and/or conversion tool again.

EEQ5027I   Renaming produced or obsolete MIF files.

Explanation: A set produced or obsolete (e.g. remaining form the previous conversion process) is being renamed to 'EEQMIFXX.MI~'.

EEQ5028E   Failed to rename file '[File1]' to '[File2]'.

Explanation: The renaming of file [File1] to [File2] failed.

User Response: Make sure the user who runs the conversion tool has sufficient access authority to rename the specified file.

EEQ5029W   Set of MIF files in '[Directory]' may not reflect a valid HCM exported configuration.

Explanation: The renaming of MIF files failed in directory [Directory] for at least one file. The reflected configuration by the remaining MIF files may not be valid.

User Response: Remove or rename produced MIF files. The set of produced MIF files may contain an invalid HCM exported configuration.

EEQ5030E   Could not process header. File cannot be processed.

Explanation: An error occurred during the processing of the header-record.

System Action: Dependent on the severity of the error the conversion process may either stop or continue.

User Response: Remove cause of the error from specified file. Make sure a valid set of HCM exported files was chosen for conversion. Make sure the HCM exported files were not changed manually.

EEQ5031W   Unknown column name. Column will be ignored.

Explanation: A unknown column-name [Name of column] was found in the header-record. This column will be ignored.

System Action: Conversion process continues. The referred column will be ignored.

User Response: Make sure that corresponding versions of HCM and conversion tool are used.

EEQ5032E   Could not process data. Current line will not be processed.

Explanation: The current data-record could not be read due to occurrence of an error.

System Action: Conversion process stops.

User Response: Remove cause of the error from specified file. Make sure a valid set of HCM exported files was chosen for conversion. Make sure the HCM exported files were not changed manually.

EEQ5050W   **Not all diagnostic messages were written to the log file. Use the 'xlog' option to get every diagnostic message.**

Explanation: There occurred more than 10 diagnostic messages and the 'log' option was given (default).

User Response: Specify the option '-xlog' to get every diagnostic message written into the log file.

EEQ5051W   **Full component name of file '[File]' too long, or current directory path too long, or source directory path too long.**

Explanation: Either the fully qualified component name of file [file], or the fully qualified name of the current directory, or the fully qualified name of the source directory is too long to be displayed.

System Action: Conversion tool stops. No conversion took place.

User Response: Shorten the fully qualified name of the specified file or directory.

EEQ5052E   **Maximum number of columns or data fields exceeded.**

Explanation: There were too much columns or data fields encountered in one row.

System Action: Conversion process stops.

User Response: Make sure that a valid set of HCM exported files was chosen for the conversion. Make sure the HCM exported files were not changed manually.

EEQ5053E   **Maximum length of [X] exceeded.**

Explanation: The maximum possible length of [X] was exceeded. where [X] = 'name of table', [X] = 'data field', or [X] = 'name of column'.

System Action: Conversion process stops.

User Response: Make sure that a valid set of HCM exported files was chosen for the conversion. Make sure the HCM exported files were not changed manually.

EEQC401E   **Host communication error**

Explanation: Invalid or expired user ID or password.

EEQC404E   **Host communication error**

Explanation: Most likely, a wrong (outdated) TP profile has been specified in the CPI-C side information.

EEQC406E   **Host communication error**

Explanation: Possible reason: APPC local node not started.

EEQC408E   **Host communication error**

Explanation: A session cannot be established for User ID user_id, symbolic destination sdn, and domain (local).

EEQC410E   **Host communication error**

Explanation: Possible reason: HCM client configuration error: name of HCM APPC host transaction program incorrectly specified, please check your CPI-C side information entry for the selected host.

EEQX100E   **Host communication error.**

Explanation: The HCM dispatcher on host <name> may be busy or not listening on port <port#>.

System Action: No connection to host.

User Response: Start the dispatcher and check you have specified the correct port number.

This message can also occur, if the HCM dispatcher has been started at a time, when TCP/IP was not already active. Stop the HCM dispatcher, start TCP/IP, and start the HCM dispatcher again. If TCP/IP is already started, stop the dispatcher and start it again.

EEQX102E   **Host communication error.**

Explanation: The hostname <name> is unknown.

System Action: No connection to host.

User Response: Verify that you spelled the hostname correctly. If the spelling is correct, use the **ping** command to check the host connection.

EEQX305E   **Host communication error.**

Explanation: Timeout waiting for HCM dispatcher reply.

System Action: No connection to host.

User Response: Increase <eeqhcm.ini[CBDQTDIS]rtimeout>. Verify that you spelled the hostname correctly.

EEQX306E   **Host communication error.**

Explanation: The message is issued because the error handling is incomplete on the dispatcher side. The login is successful, but the dispatcher fails to establish the connection between HCM and HCD. Instead of sending back an error datagram, the dispatcher just closes the socket connection. The HCM RDTS log file shows what happened.

System Action: No connection to host.

User Response: Check why the connection between HCM and HCD was disrupted. You can create the HCM log files by adding these lines to the EEQHCM.ini file:

```
[COMRDTS]
AGENTLOG=D:\ibmeeq\rdts.log
CLOGMASK=CDEFIX
[COMRHOM]
RHOMLOG=1
RHOM_EXIT_LOG=1
FORCELOG=1
CLOGMASK=CDEFIX
```

EEQX401E   Host communication error.

Explanation: Security validation failed.

System Action: No connection to host.

User Response: Check the spelling of your HCM user id.

EEQX402E   Host communication error.

Explanation: Security validation failed.

System Action: No connection to host.

User Response: Check that your password is valid.

EEQX405E   Host communication error.

Explanation: No available server port found for new HCM agent.

System Action: No connection to host.

User Response: Widen <eeqhcm.ini[CBDQAGNT]portmin..portmax>.

EEQX406E   Host communication error.

Explanation: The HCM dispatcher does not have read access to the skeleton, which is used to setup the JCL for starting the HCM agent. Or, the HCM dispatcher could not find the data set for the skeleton, which is to be used to start the HCM agent; it might not exist.

System Action: No connection to host.

User Response: Please check the data set to be used as skeleton. It should be one of:

- Default data set name (SYS1.PROCLIB(CBDQAJSK))
- Data set specified in the procedure or JCL, which has been used to start the HCM dispatcher
- Data set specified in the HCM user's ini file in the section [CBDQAGNT] with the keyword AJCLSKEL.

EEQX407E   Host communication error.

Explanation: The HCM dispatcher could not write the HCM agent JCL file.

System Action: No connection to host.

User Response: Specify [CBDQAGNT]ajclofn=<writable data set>.

EEQX410E   Host communication error.

Explanation: The HCM dispatcher could not verify a successful start of a new HCM agent.

System Action: No connection to host.

User Response: Check agent job log / agent trace on host for startup errors.

EEQX414E   Host communication error.

Explanation: The HCM dispatcher timed out waiting for a new HCM agent.

System Action: No connection to host.

User Response: Increasing <eeqhcm.ini[CBDQTDIS]atimeout> Also: check agent job log / agent trace on host for startup errors.

EEQX415E   Host communication error.

Explanation: The HCM dispatcher cannot do security validations.

System Action: No connection to host.

User Response: Verify that the dispatcher is running as an APF authorized program.

EEQX416E   Host communication error.

Explanation: The HCM dispatcher received an invalid request package.

System Action: No connection to host.

User Response: The HCM client request may have been corrupted during transfer, please retry.

EEQX500E   Host communication error.

Explanation: I/O exception waiting for HCM dispatcher reply.

System Action: No connection to host.

User Response: The connection to the HCM dispatcher was interrupted, verify that the dispatcher is still running.

EEQX501   **Host communication error.**

Explanation: Exception receiving HCM agent reply.

System Action: No connection to host.

User Response: Check if the HCM client-agent connection is lost.

EEQX502E   **Host communication error.**

Explanation: Timeout waiting for HCM agent's reply to client logon request.

System Action: No connection to host.

User Response: Increase <eeqhcm.ini[CBDQAGNT]rtimeout>.

EEQX503E   **Host communication error.**

Explanation: Timeout waiting for HCM agent reply.

System Action: No connection to host.

User Response: Check that the HCM agent is still running. Increase <eeqhcm.ini[CBDQAGNT]rtimeout>.

Appendix B. Alternate processor definitions

The physical *machine* may have multiple logical processors known as *shadows*. The *primary processor* in the processor complex is that which is in the current logical view; at IODF load/resync time, processor complex connections are built for primary processors only. *Alternate* or *shadow processors* are those not in the current view; no physical connections are built for alternate processors at IODF load/resync time. Every processor must be assigned to a new or existing processor complex. The name of the processor complex is the name of the primary processor.

You can:

* Create or delete shadow processors during an HCM session
* Confirm or change which processors are assigned to which machine during an IODF load or resync process.

Managing shadow processors during an HCM session

Display the Processors dialog by choosing the **Manage Shadow Processors...** command in the **Edit** menu. A list of logical processors, sorted by processor within processor complex, appears. You can create or delete shadow processors, change the processor style, or make a processor primary or alternate.

Creating a new processor under a machine

You can create a new processor by selecting the machine from the **Processors** dialog and pressing the **New...** button. The **Create Processor** dialog appears to allow you to create an alternate processor under the selected machine.

Deleting a processor from a machine

You can delete a processor, provided it is not the only one within the machine. Select a processor from the **Processors** dialog and press **Delete**. HCM will display a confirmation dialog before deleting the processor. Primary and alternate status settings are adjusted as required for the remaining processors in the machine.

Changing the processor style

Select a processor from the **Processors** dialog and press the **Change Style...** button. The **Change Machine Style** dialog appears, giving you the choice of an IBM style (default) or a "one processor per partition" style machine. The button is enabled if the machine contains only new processors. The button is disabled if any of the new processors involved contain partitions.

Making a processor primary or alternate

Select a processor from the **Processors** dialog and press **Primary**. For IBM style processors, the selected processor becomes the primary one. The **Primary** button is enabled only if the machine has more than one processor; it is disabled if the selected processor is already the primary one. Unselected processors within the machine become alternates.

For "one processor per partition" style processors, the button is enabled only if there is more than one processor; it is disabled if the selected processor is the only

primary processor. Otherwise, the primary/alternate state of the selected processor is toggled. Note that this style of processor may have more than one primary processor.

Assigning processors to machines during IODF load/resync

During an IODF load or resync, the **Machines** dialog will automatically appear when there are new processors (added to the IODF on the host) which must be assigned to new or existing machines, or any time that some machine has more than one processor.

If HCM discovers that the IODF being loaded contains new processors, it will attempt to assign each processor to a machine by matching first serial numbers, then SNA addresses. If unsuccessful, HCM will tentatively assign the new processor to a new machine. When the **Machine** dialog automatically appears, it will ask you to confirm or change HCM's assignments before continuing the load/resync process.

If you create a new machine to which to re-assign the new processor (by selecting the processor and pressing **New...** in the **Machine** dialog), it becomes the primary processor. If you re-assign the new processor to a different, existing machine (using the **Assign...** button to display the **Assign Machine** dialog), it becomes an alternate processor. If re-assigning the processor leaves the old machine without processors, it is deleted. Otherwise, primary and alternate settings are adjusted as necessary.

You can change a machine's style, or make it a primary or alternate processor. See "Managing shadow processors during an HCM session" on page 407.

At this point, HCM identifies new processors and labels them as 'new'. HCM also identifies deleted processors and labels them as 'deleted'.

Appendix C. "One Processor Per Partition" (OPPP) machines

An alternative to IBM's processor partitioning scheme is one in which, instead of each partition being represented by an LPAR object in the IODF, each partition is represented by its own processor. HCM calls this scheme "one processor per partition" (OPPP) and provides the following support for it.

1. IODF load and resync
2. Creating and deleting
3. Editing
4. CHPID mapping
5. Connecting
6. CTC connections

HCM represents OPPP machines similarly to the way it represents machines with multiple logical (alternate or shadow) processors. More than one processor object in the IODF can be mapped to an HCM machine. For machines with multiple logical processors, exactly one processor is designated the "primary" processor and can be edited or have connections made to it in HCM. For OPPP machines, more than one processor can be designated "primary", thus allowing you to edit and make connections to several "partitions" with HCM.

IODF load and resync

When HCM encounters new OPPP processors during IODF load or resync it maps them to existing or new machines or to its own machine by matching its serial number or SNA address. When the **Processors** dialog is opened the new processors are listed with status "NEW".

To assign an OPPP processor to another machine, select it and click **Assign**.

To change the style of a machine to OPPP, select it and click **Change Style**.

To toggle an OPPP processor between *Primary* and *Alternate*, select it and click **Primary** or **Alternate**. In OPPP machines, there can be more than one primary processor.

When the new processors have been assigned properly and the machines styles are correct, click **OK** to complete the load or resync processes.

Creating and deleting OPPP machines

To create a new OPPP machine, add a processor to an OPPP machine, or delete a processor from an OPPP machine, select the **Edit:Manage Shadow Processors** menu item.

To create a new OPPP machine, select the **Create:Processor** menu item and fill in the **Create Processor** dialog. Now select **Edit:Manage Shadow Processors**, select the new processor and click **Change Style**.

To add a new processor (partition) to an OPPP machine, select the machine in the **Shadow Processors** dialog and click **Create**. Fill in the **Create Processor** dialog to create the new processor.

To delete a processor (partition) from an OPPP machine, select the machine in the **Shadow Processors** dialog and click **Delete**.

Editing OPPP processors

To edit an OPPP processor, select the **Edit:Processor** menu item and double-click on the processor. Note the label of an OPPP machine is the same as the label of its first processor.

In the **Processor** dialog, choose the particular processor (partition) from the **Label** combo-box. Now you can edit that particular processor as you would any other processor.

CHPIDs, channels, and CHPID mapping

The distinction between CHPIDs in the IODF and channels in HCM becomes very important for OPPP machines. Each IODF processor has its own set of CHPIDs. However, the OPPP *machine* has only one set of physical *channels*.

Usually each CHPID is mapped to a channel with the same number. However this is not necessary because some OPPP machines support the ability to map CHPIDs with different numbers to the same channel or to map CHPIDs to channels beyond channel FF.

When you create or re-number CHPIDs in an OPPP processor, a *CHPID mapping* is used to determine what channel the CHPID should be mapped to. Each processor can have a different CHPID mapping. The default CHPID mapping is one-to-one between CHPID numbers and channel numbers.

To edit the CHPID mapping for an OPPP processor, edit the processor and click **CHPID Mapping** in the **Processor** dialog. This opens the **CHPID <> Channel Mapping** dialog.
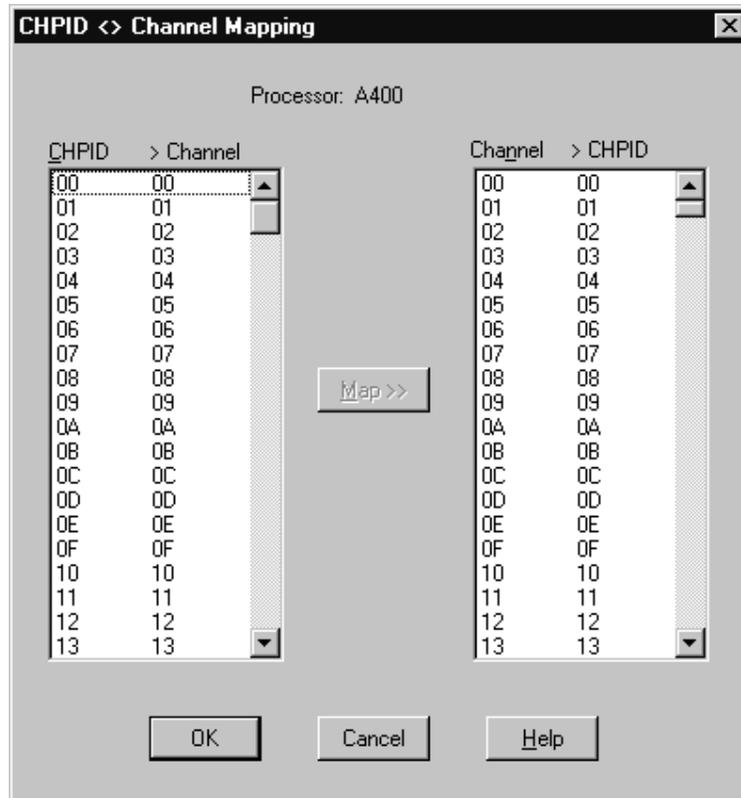
Figure 163. CHPID <> Channel Mapping Dialog

The listbox on the left lists all possible CHPID numbers (00- FF) and the channel numbers to which they map. The listbox on the right lists all possible channel numbers (00-1FF) and the CHPID numbers to which they map.

To change the mapping for a range of CHPID numbers, select the range in the left listbox and select the first channel number in the right listbox. Click **Map** to map the selected range of CHPID numbers to as many of channel numbers starting with the selected channel number.

If any of the target channel numbers is already mapped, the previous mapping is discarded. The previous CHPID numbers must be re-mapped before you can close this dialog (unless you press Cancel.)

After changing the CHPID mapping, any existing CHPIDs are remapped as appropriate.

Connecting OPPP machines

When connecting to a channel on an OPPP machine, the connection is made to all CHPIDs that are mapped to that channel, and therefore also to the processors of those CHPIDs.

CTC connections to OPPP machines

When making a CTC connection to a channel on an OPPP machine that has more than one CHPID, HCM will prompt you to specify which CHPID (and therefore which processor) to make the connection to. To make CTC connections to several CHPIDs that map to one channel, follow the procedure for making a CTC connection once for each CHPID.

Appendix D. Naming conventions in dialogs and reports

Objects in HCM are usually described in dialogs, reports, on the configuration diagram, in the status bar, in the object information bar, and on certain menu items using a standard format. The format used for most situations is described below.

Object	Formatting convention (NAME)	Examples
Processor	ProcessorName	PROD1
Partition	ProcessorName.PartitionName	PROD1.LPARA
CHPID	ProcessorName.CHPID	PROD1.2F
Switch	SwitchName	ESCD_1
Switch Port	SwitchName.SwitchPortID	ESCD_1.E3
Crossbar	CrossbarName	PRN-XBAR
Crossbar Interface	CrossbarName.CrossbarInterfaceName	PRN-XBAR.1 (channel interface) PRN-XBAR.A1 (CU interface)
Controller	ControllerName	3990 #5
Controller Segment	ControllerName.SegmentName	3990 #5.SC0 "SEG" if none assigned by PDF
Controller (Channel / Device) Adapter	ControllerName.SegmentName.AdapterName"Segment Name" is omitted if none assigned by PDF "AdapterName" is "CA" "DA" if none assigned by PDF.	6110 #1.Bus 0.A (CA) 6110 #1.Bus 0.0 (DA)
Controller Interface	ControllerName.SegmentName.AdapterName.InterfaceName "INT" if none assigned by PDF	3990 #5.SC0.A 3990 #5.SC0.SP0 6110 #1.Bus 0.A.0 6110 #1.Bus 0.0.INT
String	StringName "String" plus range of device addresses if none assigned	BACKUP DASD String 500-51F
Unit	StringName.UnitName "Unit" plus range of device addresses if none assigned.	String 500-51F.Unit 500-507
Device	DeviceNumber or DeviceNumber,Count	0500 0500,8
String Interface	StringName.InterfaceName "INT" if none assigned by PDF	String 500-51F.A
Converter	ConverterName "S->P" or "P->S" if no label assigned	S->P between TEST1.22 and 3990 #1.SC0.C
Cabinet	CabinetName	Central Cabinet
Patchport	PortId	C101

In some situations, the formatted NAME as described above may be followed by additional information. The exact information that is included varies with circumstances.

Object	Formatting convention (NAME plus additional information)	Examples
Processor	ProcessorName TypeModel "Description" SN:Serial Number	PROD1 9021-972 "Production 1" SN: 1234569021
Partition (IBM style machine)	ProcessorName.PartitionName ImageNumber Usage "Description"	PROD1.LPARA 01 OS "OS Partition #1"
Partition (one processor per partition style machine)	ProcessorName.PartitionName TypeModel "Description"	PROD1.DOM1.3090 "Domain 1"
CHPID	ProcessorName.CHPID InterfaceType Type Mode "Description" Connections	PROD1.21 S CNC SHR "DIR 2.E8" to ESCD_2.E8
CHPID with no physical channel	ProcessorName.CHPID (No Channel) Type Mode "Description"	PROD1.21 (No Channel) ""
Physical channel with no CHPID	ProcessorName.CHPID InterfaceType (No CHPID) Connections	PROD1.21 S (No CHPID) to ESCD_2.E8
CHPID on one processor per partition machine	ProcessorName.CHPID InterfaceType (n CHPIDs) Connections	PROD1.21 S (3 CHPIDs) to ESCD_2.E8
Switch	SwitchName UnitModel "Description" SN:SerialNumber	ESCD_2 9032 "DASD Switch" SN:79023454
Switch Port	SwitchName.SwitchPortID LASER/LED "Name" Connections	ESCD_2.E8 LED "P1.21" from PROD1.21
Crossbar	CrossbarName SN:SerialNumber	PRN_XBAR SN: 0987242
Crossbar Interface	CrossbarName.CrossbarInterfaceName Connections	PRN_XBAR.A1 to 4245 #1.A
Controller	ControllerName SN:SerialNumber Strings	3990 #5 SN:98734789 reaching String 400-41F String 420-43F
Controller Segment	ControllerName.SegmentName Strings	3990 #5.SC0 reaching String 400-41F String 420-43F
Controller Interface	ControllerName.SegmentName.AdapterName.InterfaceName InterfaceType Connections	3990 #5.SC0.A S from ESCD_1.CD
String	StringName UnitModel Controllers "UnitModel" - taken from first device	String 500-51F 3390 on 3990 #5
Unit	StringName.UnitName UnitModel SN:SerialNumber "UnitModel" - taken from first device	String 500-51F.Unit 500-507 3390 SN:74329874
Device	DeviceNumber VOLSER	0500 489327
String Interface	StringName.InterfaceName Connections	String 500-51F.A from 3990 #5.SC0.SP0
Convertor	ConvertorName SN:SerialNumber Connections	P->S 293473284 between PROD1.06 and 3990 #11.SC1.A
Cabinet	CabinetName SN:SerialNumber	Central Cabinet 077349782

Cable descriptions

On the diagram, cable information will be in one of the following formats, depending on the zoom level:

Number of Lines showing	3	2	1
Short name and interface:	PR-54	PR-54	PR-54
Cable Identification:	5676	5676	
Cable length:	100		

Short name is the two letter abbreviation of the major object that the cable connects to.

In the above example; 100 is the length of the cable in unspecified units, 5676 is the cable identification and PR-54 identifies the cable connected to CHPID number 54 on the processor with the short name "PR".

Appendix E. Menu commands summary

The menu choices are acting either on the *configuration work (*.HCM)* file or on the *configuration production (*.HCR)* file.

File menu

Open...	Open an existing configuration (.HCM, .HCR) file and the corresponding IODF.
Open Specia**l**...	Open an existing configuration (.HCM, .HCR) file and a specified IODF.
Save	Save the current configuration (.HCM, .HCR) file with a timestamp that is synchronized with the IODF, and continue the session with the files open.
Close	Save the configuration (.HCM, .HCR) file and the corresponding IODF, and then close them.
IODF **D**etails...	Display important information about the IODF.
Copy to **B**ackup...	Copy the configuration (.HCM, .HCR) file to a new file and the corresponding IODF to a new data set.
Build **P**roduction...	Create a production IODF on the host out of the currently open work IODF, while at the same time creating a production (*.HCR) file out of the currently open work (*.HCM) file on the PWS.
	If the IODF contains XMP processors where at least one PCHID value is missing, but else is complete and valid, the work IODF is transformed into a validated work IODF. For more information on how to proceed with validated work IODFs, refer to "Making a work or a production IODF" on page 83.
Make **W**ork...	Create a work IODF on the host out of the currently open production IODF, while at the same time creating a work (*.HCM) file out of the currently open production (*.HCR) file on the PWS.
IODFs...	Open a dialog to manage IODFs.
Page Se**t**up...	Setup the page for printing purposes; setup the printer and its parameters.
Print Dia**g**ram...	Print the configuration diagram.
Print **R**eports...	Print reports and context diagrams.
Print Cable L**a**bels...	Print cable labels to a text file or printer.
I**m**port Data...	Import new and changed non-HCM data into an existing or a new HCM configuration.
Export Data...	Export configuration data to a text file in a directory.
Ex**i**t (Alt+F4)	Save and close the configuration (.HCM, .HCR) file and the corresponding IODF and exit HCM.

Edit menu

Undo last change (Alt+Backspace)	Undo the last change made to the configuration. This menu item's text specifies the action that is currently available to be undone. If the previous action cannot be undone, the item's text changes to "Can't Undo".

Delete/disconnect selected object (Del)	Delete the most recently selected object in the configuration diagram. This menu item's text changes to describe the most recently selected object. Shortcut: press the DEL key after selecting the object.
Edit selected object... (F9)	Edit the most recently selected object. This menu item's text changes to describe the most recently selected object. Shortcut: double-click the object in the diagram, or click on the object and press F9.
Edit cable on selected interface (F5)	Edit the cable on the most recently selected object. This menu item's text changes to describe the most recently selected object.
Processor...	Edit a particular processor.
Partition...	Edit a particular partition.
CHPID...	Edit a particular CHPID.
Switch...	Edit a particular switch.
Port...	Edit a particular port.
Crossbar Switch...	Edit a particular crossbar switch.
Controller...	Edit a particular controller.
Control Unit...	Edit the controller containing a particular control unit.
String...	Edit a particular string.
Device...	Edit a particular device.
Converter...	Edit a particular converter.
Patchport...	Edit the cabinet containing a particular patchport.
Cabinet...	Edit a particular cabinet.
Cable Assignments...	Edit the cable assignments to connections.
OS Configurations...	Edit the OS configurations.
CTC connections...	Edit CTC connections.
CF connections...	Edit CF connections.
PPRC connections...	Edit Point-to-Point connections (PPRC connections).
Order Processors...	Change the order of processors in the configuration diagram.
Titles & Footnotes...	Edit the title and footnotes that appear in configuration diagrams and reports.
Manage Shadow Processors...	Edit the assignment of alternate (shadow) processors to physical machines.

View menu

Previous View (F12)	Display the previous automatically saved view of a configuration diagram.
Fit to Window (Ctrl+Home or F2)	Reduce the scale of the configuration diagram to fill the window.
Zoom Out (Home)	Reduce the scale of the configuration diagram by one step.
Zoom In (End)	Enlarge the scale of the configuration diagram by one step.
Maximum Size (Ctrl+End)	Enlarge the scale of the configuration diagram to the maximum size
Filter Diagram...	Specify which objects are included in the diagram.
Connected Objects (Shift+F4)	Crop the view to display only objects logically connected to the selected object. Settings in the *Change View Attributes* dialog are not considered.
Highlighted Objects (F4)	Crop the view to display only objects logically connected to the selected object. Settings in the *Change View Attributes* dialog are considered.
Visualized Matrices	Define a configuration for each ESCON or FICON switch or crossbar switch to be visualized.
Restore Named View (F3)	Select a saved named view to be shown as active configuration diagram.
Save Named View	Save the displayed view.

<u>D</u>elete Named View	Delete a named view.
<u>R</u>estore Matrix Set	Select a saved named matrix view to be shown in the configuration diagram.
<u>S</u>ave Matrix Set	Save the current matrix set.
<u>D</u>elete Matrix Set	Delete a named matrix set.
<u>T</u>ext Detail...	Specify which text appears in the configuration diagram.
Preferen<u>c</u>es...	Set certain preferences for the configuration diagram.
Colors...	Enable/disable and set colors for visual warnings for logical/physical inconsistencies and for status information.
Ob<u>j</u>ect Information Bar	Include or exclude the Object Information Bar.
Sta<u>tu</u>s Bar	Include or exclude the Status Bar.
Show Page <u>B</u>reaks	Display the effects of page breaks and printer margins.

Locate menu

Se<u>l</u>ected object (F8)	Locate the most recently selected object. This menu item's text changes to describe the most recently selected object.
S<u>y</u>stem	Locate a system in a sysplex. This menu item's text changes to describe the most recently selected object.
<u>P</u>rocessor...	Locate a particular processor
P<u>a</u>rtition...	Locate a particular partition
C<u>H</u>PID...	Locate a particular CHPID
Sw<u>i</u>tch...	Locate a particular switch
P<u>o</u>rt...	Edit a particular port.
Crossbar S<u>w</u>itch...	Locate a particular crossbar switch
<u>C</u>ontroller...	Locate a particular controller.
<u>C</u>ontrol <u>U</u>nit...	Locate the controller containing a particular control unit.
<u>S</u>tring...	Locate a particular string.
<u>D</u>evice...	Locate a particular device.
Con<u>v</u>erter...	Locate a particular converter.
Patchp<u>o</u>rt...	Locate the cabinet containing a particular patchport.
Cabi<u>n</u>et...	Locate a particular cabinet.
C<u>T</u>C connection...	Locate a particular CTC connection.
C<u>F</u> connection...	Locate a particular CF connection.
PPRC conn<u>e</u>ction...	Locate a particular Point-to-Point connection (PPRC connection).
Ca<u>b</u>le...	Locate a particular cable.

Create menu

<u>P</u>rocessor...	Create a new processor.
Sw<u>i</u>tch...	Create a new switch.
Crossbar S<u>w</u>itch...	Create a new crossbar switch.
<u>C</u>ontroller...	Create a new controller.
<u>S</u>tring...	Create a new string.
Cabi<u>n</u>et...	Create a new cabinet.
C<u>T</u>C connection...	Create a new CTC connection.
C<u>F</u> connection...	Create a new CF connection.
PPRC conn<u>e</u>ction...	Create a Point-to-Point connection (PPRC connection).

Operations menu

Refresch	Obtain system status data
Sense	Obtain "sensed" data from the host
Focus on host...	Select the host for which the status information is to be displayed
Next focus on host	Jump to the next host as the focus host
Comand...	Opens the I/O Operations command dialog window
Preferences...	Set preferences specific to I/O Operations

Utilities

Move Port Connections...	Move port connections from a switch to another or within a single switch.
Aggregate CHPIDs...	Aggregate several CHPID definitions onto one single CHPID.
Convert ESCON Port to **F**C Port...	Migrate CNC channels to fiber channels.
Prime the IODF with Sensed Data...	Write sensed serial numbers and VOLSERs into the IODF.
Change CHPID **L**ink Address Format...	Create a PPRC connection consisting of up to eight single PPRC links, and provide parameters for the PPRC connection.
Create **S**ubsystem...	Create an I/O subsystem consisting of controllers (with segments and control units) and strings (with devices).

Options menu

Confirmation...	Instruct HCM whether to prompt for a confirmation before deleting an object or disconnecting an interface.
Field Names...	Assign customized names to the user data fields.
Interface Selection...	Choose whether to select the object containing an interface or its cable when an interface is selected with the mouse.
Activate HCD **T**race	Turn the HCD trace facility on or off.

Help menu

Index	Display the main HCM help topic index.
Keyboard	Describe the use of the Function keys and various cursor control key shortcuts.
Using **H**elp	Open an index of topics that describe the Windows Help facility.
About...	Open a window describing the current release of HCM.

Appendix F. Table description for Tivoli Inventory Support

The following chapter describes the tables that were added to the Tivoli Inventory database in order to store the HCM exported data. The tables are also stored in the product directory [Drive]:\IBMEEQ\MIFUTIL\TABLDESC.TXT. The data type of all fields is VARCHAR. For the content of the fields refer to the description of the HCM exported data in "Table summary" on page 333.

Table name : **HCMHWPROC**
MIF-file name : EEQMIF00.MIF
HCM exported file : PROC.TXT

Column name	
PROC$ID	PROC$MODE
PROC$GROUPED$WITH	PROC$NETWORK$NAME
PROC$STYLE	PROCCPCNAME
PROC$STATUS	PROC$USER$FIELD1
PROC$SHORT$NAME	PROC$USER$FIELD2
PROC$SERIAL$NUMBER	PROC$USER$FIELD3
PROC$DESCRIPTION	PROC$USER$FIELD4
PROC$MODEL	PROC$USER$FIELD5
PROC$SUPPORT$LEVEL	

Table name : **HCMHWPART**
MIF-file name : EEQMIF01.MIF
HCM exported file : PART.TXT

Column name	
PART$ID	PART$USER$FIELD2
PART$DESCRIPTION	PART$USER$FIELD3
PART$NUMBER	PART$USER$FIELD4
PART$USAGE	PART$USER$FIELD5
PART$USER$FIELD1	

Table name : **HCMHWCHPID**
MIF-file name : EEQMIF02.MIF
HCM exported file : CHPID.TXT

Column name	
CHPID$ID	CHPIDDYNSWITCH
CHPID$UNIQUE$KEY	CHPID$PHYS$NUMBER
CHPID$DESCRIPTION	CHPIDINTTYPE
CHPID$TYPE	CHPID$PCHID
CHPID$MODE	CHPID$SPANNED$WITH

Table name : **HCMHWDIR**
MIF-file name : EEQMIF03.MIF
HCM exported file : DIR.TXT

Column name	
DIR$ID	DIR$USER$FIELD1
DIR$SERIAL$NUMBER	DIR$USER$FIELD2
DIR$DESCRIPTION	DIR$USER$FIELD3
DIR$MODEL	DIR$USER$FIELD4
DIR$CONTR$LABEL	DIE$USER$FIELD5
DIR$STRING$LABEL	

Table name : **HCMHWPORT**

MIF-file name : EEQMIF04.MIF

HCM exported file : PORT.TXT

Column name	
PORT$ID	PORTINTTYPE
PORT$UNIQUE$KEY	PORT$NAME

Table name : **HCM$HWCONTRLLR**

MIF-file name : EEQMIF05.MIF

HCM exported file : CONTRLLR.TXT

Column name	
CONTR$LABEL	CONTR$INTEGRATED
CONTRSERNUM	CONTR$GROUPED$WITH
CONTRDEVTYPE	CONTR$USER$FIELD1
CONTR$IODF$CUT (CU type)	CONTR$USER$FIELD2
CONTRPDFABID (abbrevid)	CONTR$USER$FIELD3
CONTRPDFNAME	CONTR$USER$FIELD4
CONTR$DISP$CLASS	CONTR$USER$FIELD5
CONTR$INSTALLED	

Table name : **HCMHWCCI**

MIF-file name : EEQMIF06.MIF

HCM exported file : CCI.TXT

Column name	
CCIINTNAME	CCI$INTERFACE$TYPE
CCI$UNIQUE$KEY	

Table name : **HCMHWCDI**

MIF-file name : EEQMIF07.MIF

HCM exported file : CDI.TXT

Column name	
CDIINTNAME	CDI$UNIQUE$KEY

Table name : **HCMHWCU**

MIF-file name : EEQMIF08.MIF

HCM exported file : CU.TXT

Column name

CU$ID	CU$DESCRIPTION
CU$CONTAINER	CU$TYPE
CU$SERIAL$NUMBER	

Table name : **HCMHWSTRING**

MIF-file name : EEQMIF09.MIF

HCM exported file : STRING.TXT

Column name

STRING$TAG	STRING$USER$FIELD1
STRING$IODF$DT (device type)	STRING$USER$FIELD2
STRINGPDFABID (abbrevid)	STRING$USER$FIELD3
STRINGPDFNAME	STRING$USER$FIELD4
STRING$LABEL	STRING$USER$FIELD5
STRING$INSTALLED	

Table name : **HCMHWSI**

MIF-file name : EEQMIF0A.MIF

HCM exported file : SI.TXT

Column name

SIINTNAME	SI$UNIQUE$KEY

Table name : **HCMHWUNIT**

MIF-file name : EEQMIF0B.MIF

HCM exported file : UNIT.TXT

Column name

UNIT$TAG	UNIT$USER$FIELD2
UNIT$SERIAL$NUMBER	UNIT$USER$FIELD3
UNIT$LABEL	UNIT$USER$FIELD4
UNIT$INSTALLED	UNIT$USER$FIELD5
UNIT$USER$FIELD1	

Table name : **HCMHWDEVICE**

MIF-file name : EEQMIF0C.MIF

HCM exported file : DEVICE.TXT

Column name

DEVICE$ID	DEVICE$DESCRP
DEVICE$CONTAINER	DEVICE$VOLSER
DEVICESERNUMBER	DEVICE$TYPE

Table name : **HCMHWXBAR**

MIF-file name : EEQMIF0D.MIF

HCM exported file : XBAR.TXT

Column name

XBAR$LABEL	XBAR$USER$FIELD1
XBAR$NUMOF$CI	XBAR$USER$FIELD2
XBAR$NUMOF$CUI	XBAR$USER$FIELD3
XBAR$SERIAL$NUMBER	XBAR$USER$FIELD4
XBAR$SHORT$NAME	XBAR$USER$FIELD5
XBAR$DEFAULT$CFG	

Table name : **HCMHWXBARCFG**

MIF-file name : EEQMIF0E.MIF

HCM exported file : XBARCFG.TXT

Column name

XBARCFG$NAME	XBARCFG$ROW17$CI
XBARCFG$ROW1$CI	XBARCFG$ROW18$CI
XBARCFG$ROW2$CI	XBARCFG$ROW19$CI
XBARCFG$ROW3$CI	XBARCFG$ROW20$CI
XBARCFG$ROW4$CI	XBARCFG$ROW21$CI
XBARCFG$ROW5$CI	XBARCFG$ROW22$CI
XBARCFG$ROW6$CI	XBARCFG$ROW23$CI
XBARCFG$ROW7$CI	XBARCFG$ROW24$CI
XBARCFG$ROW8$CI	XBARCFG$ROW25$CI
XBARCFG$ROW9$CI	XBARCFG$ROW26$CI
XBARCFG$ROW10$CI	XBARCFG$ROW27$CI
XBARCFG$ROW11$CI	XBARCFG$ROW28$CI
XBARCFG$ROW12$CI	XBARCFG$ROW29$CI
XBARCFG$ROW13$CI	XBARCFG$ROW30$CI
XBARCFG$ROW14$CI	XBARCFG$ROW31$CI
XBARCFG$ROW15$CI	XBARCFG$ROW32$CI
XBARCFG$ROW16$CI	

Table name : **HCMHWXCI**

MIF-file name : EEQMIF0F.MIF

HCM exported file : XCI.TXT

Column name

XCIINTNAME	XCI$UNIQUE$KEY

Table name : **HCMHWXCUI**

MIF-file name : EEQMIF10.MIF

HCM exported file : XCUI.TXT

Column name

XCUIINTNAME	XCUI$UNIQUE$KEY

Table name : **HCMHWCABINET**	
MIF-file name : EEQMIF11.MIF	
HCM exported file : CABINET.TXT	
Column name	
CABINET$ID	CABINETUSFLD1
CABINET$DESCRIPTION	CABINETUSFLD2
CABINET$MODEL	CABINETUSFLD3
CABINETSERNUM	CABINETUSFLD4
CABINETGRPPOS	CABINETUSFLD5

Table name : **HCMHWPATCHPRT**	
MIF-file name : EEQMIF12.MIF	
HCM exported file : PATCHPRT.TXT	
Column name	
PATCHPRT$NAME	PATCHPRTEXLAB
PATCHPRTUNKEY	PATCHPRTINST
PATCHPRT$CABINET	

Table name : **HCMHWCONVERTR**	
MIF-file name : EEQMIF13.MIF	
HCM exported file : CONVERTR.TXT	
Column name	
CONVERTR$LABEL	CONVERTRUSFLD1
CONVERTRUNKEY	CONVERTRUSFLD2
CONVERTR$TYPE	CONVERTRUSFLD3
CONVERTR$POS	CONVERTRUSFLD4
CONVERTRSERNUM	CONVERTRUSFLD5

Table name : **HCMHWCABLE**	
MIF-file name : EEQMIF14.MIF	
HCM exported file : CABLE.TXT	
Column name	
CABLE$ID	CABLE$LOSS$A
CABLE$TYPE	CABLE$LOSS$B
CABLE$LENGTH	CABLE$COMMENT

Table name : **HCMHWLOGLINK**	
MIF-file name : EEQMIF15.MIF	
HCM exported file : LOGLINK.TXT	
Column name	
LOGLINK$SOURCE	LOGLINKTARTYPE
LOGLINKSRCTYPE	LOGLINKTARUK
LOGLINKSRCUK	LOGLINKTRMLOSS
LOGLINK$TARGET	LOGLINKRECLOSS

Table name : **HCMHWCONNECT**
MIF-file name : EEQMIF16.MIF
HCM exported file : CONNECT.TXT
Column name

CONNECT$SOURCE	CONNECTTARUK
CONNECTSRCTYPE	CONNECT$LLTAR$UK
CONNECTSRCUK	CONNECT$SEQUENCE
CONNECT$TARGET	CONNECT$CABLE$ID
CONNECTTARTYPE	CONNECT$PERMANENT

Table name : **HCMHWDEVCU**
MIF-file name : EEQMIF17.MIF
HCM exported file : DEVCU.TXT
Column name

DEVCU$DEVICE$ID	DEVCUCUID

Table name : **HCMHWCUCHPID**
MIF-file name : EEQMIF18.MIF
HCM exported file : CUCHPID.TXT
Column name

CUCHPIDCUID	CUCHPID$DLA
CUCHPIDCHPID	CUCHPIDDLASET

Table name : **HCMHWDEVIMAGE**
MIF-file name : EEQMIF19.MIF
HCM exported file : DEVIMAGE.TXT
Column name

DEVIMAGEDEVID	DEVIMAGELNTYPE
DEVIMAGEIMID	

Table name : **HCMHWPANEL**
MIF-file name : EEQMIF1A.MIF
HCM exported file : PANEL.TXT
Column name

PANEL$ID	PANEL$NUMOF$ROW
PANEL$DESC	PANEL$NUMOF$COL
PANELNAMFORM	PANELSR1ST

Table name : **HCMHWPPRCLINK**
MIF-file name : EEQMIF1B.MIF
HCM exported file : PPRCLINK.TXT
Column name
PPRCLINK$SOURCE$IF   PPRCLINK$DESC PPRCLINK$SOURCE$CU   PPRCLINK$DEVN PPRCLINK$TARGET$CU   PPRCLINK$PRIM$SSID PPRCLINK$LINK$ADDR   PPRCLINKSECSSID PPRCLINK$CUADD   PPRCLINK$LINK

Appendix G. Special setup considerations for z/OS

The following allows special individual adaptation for setup. In general, it is not necessary to use these parameters. They are provided for special purposes.

Usually, only AJCLACCT or RTIMEOUT for [CBDQAGNT] are of interest for common purposes. The other keywords are mainly for debugging purposes.

HCM client: configurable login parameters

The only login parameter needed is the TCP/IP hostname of the MVS host running the HCM dispatcher. If no port is specified (port = 0), the client will use the pre-configured default port number 51107 to contact the HCM dispatcher.

During HCM login, the HCM client reads the HCM Windows eeqhcm.ini file, which resides in the Windows system directory (e.g., c:\winnt\eeqhcm.ini) to access optional parameters for the login request.

The following parameters can be defined for the sections [COMRDTS], [CBDQTDIS] and [CBDQAGNT]:

[COMRDTS] AGENTLOG=<path-of-client-logfile>
> This activates the writing of an HCM client log to the specified path and filename. Directories specified in the path must already exist. The client log can be helpful in diagnosing HCM login problems.
>
> Example: AGENTLOG=c:\hcm\client.log

[CBDQTDIS] RTIMEOUT=<seconds>
> This defines the number of seconds the HCM client waits for an answer from the HCM dispatcher after sending an HCM login request and before reporting a timeout error. The default is RTIMEOUT=90. On slow networks or when the priority of the dispatcher is low, this timeout value may have to be increased.

[CBDQTDIS] ATIMEOUT=<seconds>
> The HCM dispatcher starts a new HCM agent by writing the agent job to the JES internal reader. It then attempts to contact the new agent to verify successful startup. The ATIMEOUT parameter specifies the number of seconds the dispatcher will wait for the agent to start before reporting a failure to the waiting HCM client. The default is ATIMEOUT=60. If the JES input queue is slow in starting the submitted agent job, this value may have to be increased.

[CBDQAGNT] AJCLSKEL=<HCM agent skeleton JCL data set>
> This overwrites the default JCL skeleton used by the HCM dispatcher to generate the HCM agent job. This can be used, for example, to start an HCM agent with an experimental version of the HCDPROF data set (instead of the default &SYSUID..HCD.PROFILE).
>
> Fully qualified MVS data set names must be specified within two pairs of single quotes, for example: AJCLSKEL=''TEST.CBDQAGNT.JST''

[CBDQAGNT] AJCLASS=<job class of HCM agent>
> This overwrites the default job class (0) of the generated HCM agent job. It can be used to save the job output of the HCM agent by specifying, for example, 'AJCLASS=A'.

[CBDQAGNT] AMCLASS=<agent job message class>
> This defines the message class parameter in the JOB statement of the generated agent JCL. The default message class is "0". It may be necessary to specify a different job message class in order to obtain a job log for the HCM agent. Example: AMCLASS=H

[CBDQAGNT] RTIMEOUT=<seconds>
> This defines the number of seconds the HCM client waits for an answer from the HCM agent after sending a request and before reporting a timeout error. The default is RTIMEOUT=600. For long-running HCD requests, this timeout value may have to be increased.

[CBDQAGNT] ATIMEOUT=<seconds>
> When a new HCM agent is started, it is first contacted by the HCM dispatcher. After sending a ready notification to the dispatcher, the agent then waits to be contacted by the HCM client. If the agent is not contacted by the client after the specified timeout value (e.g., the HCM user accidentally hits power-off on the PC), the agent will terminate. The default is ATIMEOUT=60. On slow networks, this value may have to be increased.

[CBDQAGNT] AJCLACCT=<accounting information>
> This defines the accounting information parameter in the JOB statement of the generated agent JCL. If the installation requires accounting information, it may be necessary to supply this parameter if the default accounting information supplied by the HCM dispatcher is invalid for the individual HCM user. Example: AJCLACCT=(DE03160,,)

[CBDQAGNT] AREGION=<agent job region size>
> This defines the region parameter in the JOB statement of the generated agent JCL. The default region size is AREGION=128M.

[CBDQAGNT] ALOGDSN=<name of agent log data set>
> This defines the name of a log data set written by the HCM agent. The default is that no log output is written. Log output may be helpful in diagnosing communication problems between the HCM client and the HCM agent.

> **Example 1: ALOGDSN=CBDQAGNT.LOG**
>> Writes the agent log as CBDQAGNT.LOG under the user's HLQ

> **Example 2: ALOGDSN="TEMP.CBDQAGNT.LOG"**
>> Writes the agent log as 'TEMP.CBDQAGNT.LOG' (fully qualified)

> **Example 3: ALOGDSN=DD:SYSOUT**
>> Writes the agent log to the job log output

[CBDQAGNT] HCDPROF=<DD specification of HCD profile data set>
> This defines the DD specification of the HCD profile used by the HCM agent. The default is:
> ```
> HCDPROF="//HCDPROF  DD DISP=SHR,DSN=&SYSUID..HCD.PROFILE"
> ```
>
> Specifying this parameter allows the use of an alternate HCD profile, e.g.,
> ```
> HCDPROF="//HCDPROF  DD DISP=SHR,DSN=&SYSUID..MYHCD01.PROFILE"
> ```

[CBDQAGNT] HCDTRACE=<DD specification of HCD trace data set>
> This defines the DD specification of the HCD trace data set used by the
> HCM agent. The default is:
>
> ```
> HCDTRACE="//HCDTRACE DD DISP=SHR,DSN=&SYSUID..HCM.TRACE"
> ```
>
> Specifying this parameter allows the use of an alternate HCD trace data
> set, e.g.,
>
> ```
> HCDTRACE="//HCDTRACE DD DISP=SHR,DSN=&SYSUID..MYHCM01.TRACE"
> ```

[CBDQAGNT] ASTEPLIB=<name of steplib data set>
> This defines a data set name which is appended to the STEPLIB
> concatenation. Per default, there is no additional steplib data set.
>
> Example: ASTEPLIB=BMGN.MY.LOADLIB

HCM INI file: EEQHCM.INI

> The HCM INI file (EEQHCM.INI) is created after the first successful HCM
> session. Usually, this is sufficient. There might be situations, however, when
> you need an INI file for the first login request. In this case, you can simply
> use an editor and create the INI file manually with the desired contents. For
> example (c:\winnt\eeqhcm.ini):
>
> ```
> [HCM]
> ERRLOGDIR=c:\HCMLOG
> [CBDQAGNT]
> RTIMEOUT=6000
> AJCLACCT=(DE03160,,)
> [COMRDTS]
> AGENTLOG=c:\hcmlog\tstLOG.LOG
> ```
>
> **Note:** If you get timeout messages (like EEQX305E or EEQX414E) when trying to
> establish an HCM-HCD session for the first time, increasing timeout values
> in the ini file might not help, because the the defaults are usually sufficient.
> The problem might lie in the TCP/IP configuration/customization (LE
> environment, DNS name server, etc.). First check the reason for the timeout
> before increasing the defaults. See Chapter 18, "Problem determination," on
> page 381 for more details.

Host messages

> The following messages can be issued by either the HCM dispatcher or the HCM
> agent on the host. Theses messages will be written to the SDSF console (issued as a
> WTO message).

CBDG980I CBDQTDIS -- HCM dispatcher starting, port = <port>
> The HCM dispatcher is starting and will listen for start-agent requests on
> the specified TCP/IP port.

CBDG981E CBDQTDIS -- can't listen on port <port>
> The HCM dispatcher cannot bind to the specified port and will terminate.
> The port number may be in use by another program.
>
> Restart the HCM dispatcher, possibly specifying a different (available) port
> number.

CBDG982I CBDQTDIS -- HCM dispatcher terminating.
> The HCM dispatcher is terminating.

CBDG983I CBDQTDIS -- start-agent request for user <user_id> received.
The HCM dispatcher received an HCM start-agent request from a client for the specified user ID.

CBDG984I CBDQTDIS -- HCM agent started for user <user_id>, port <port>.
The HCM dispatcher successfully started an HCM agent for the specified client user ID. The new HCM agent instance listens on the specified TCP/IP port number.

CBDG985I CBDQTDIS -- HCM validation for client user id <user_id> failed.
The HCM dispatcher could not successfully validate the HCM start-agent request for the specified user. Either the dispatcher is not running as an authorized program or the login information specified by the client is invalid.

For operators: make sure that the HCM dispatcher is being run from an authorized library.

For users: check the spelling of your user ID and password. Also check whether your TSO password has expired. Then retry the HCM client login.

CBDG986I CBDQAGNT -- HCM agent starting, port = <port>
An instance of the HCM agent was started successfully and is expecting HCM requests on the specified TCP/IP port number.

CBDG987I <module_name> <variable_text>
The text of this message provides additional information for an already issued message (CBDG980I to CBDQG986I). The content of this message depends on the situation.

HCM Client: TCP/IP related messages

Messages indicating a problem related to the TCP/IP communication between the HCM client on the workstation and the HCD server on the host have the message identifiers EEQX100E to EEQX503E. For the message text see Appendix A, "HCM messages," on page 393.

Appendix H. Accessibility

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 Volume 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/Library Server versions of z/OS books in the Internet library at:

http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

One exception is command syntax that is published in railroad track format; screen-readable copies of z/OS books with that syntax information are separately available in HTML zipped file form upon request to mhvrcfs@us.ibm.com.

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

> IBM Corporation
> Mail Station P300
> 522 South Road
> Poughkeepsie New York 12601-5400
> U.S.A.

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 document 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.

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.

If you are viewing this information in softcopy form, photographs and color illustrations may not appear.

Trademarks

The following terms, used in this publication, are trademarks of International Business Machines Corporation in the United States and/or other countries:

1-2-3	OS/390
3090	Parallel Sysplex
AS/400	RACF
BookManager	RAMAC
DB2	RAMAC
Enterprise Storage Server	Resource Link
eServer	SP
ES/9000	S/390
ESCON	Tivoli
FICON	VTAM
GDPS	Tivoli
IBM	VTAM
Lotus	z/OS
MVS	zSeries
MVS/ESA	z/VM

Microsoft, Windows, Windows NT, and Windows 2000 are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Pentium is a registered trademark of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Accessing licensed books on the web (z/OS)

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

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

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web user ID and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web user ID and password log on to:

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

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

Note: You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

Accessing licensed books on the web (z/VM)

The latest editions of most z/VM publications are available in Adobe Acrobat PDF format or IBM BookManager® format from the IBM VM Internet Library:

```
http://www.ibm.com/eserver/zseries/zvm/library/
```

The IBM VM Internet Library also includes other information about z/VM, such as:

- Program directories
- Data areas and control blocks
- Monitor records

Using LookAt system for online message lookup

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some codes. LookAt now includes some z/VM system messages and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

```
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html
```

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS). For those using z/VM, use it as you would any other z/VM command.

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, IAT1836 or IAT*). You can select a specific release to narrow your search. You can also download code from the *z/OS Collection*, SK3T-4269 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO or z/VM command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO and z/VM (it is the same code) from a disk on your *z/OS Collection*, SK3T-4269 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM.**
4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO or z/VM command line, simply enter: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For such messages, LookAt prompts you to choose which book to open.

Glossary

This glossary defines technical terms and abbreviations used in the Hardware Configuration Manager documentation. If you do not find the term you are looking for, refer to the index of the appropriate HCD manual or view the *IBM Glossary of Computing Terms*, available from: **http://www.ibm.com/ibm/terminology**.

A

access list. A CHPID has two partition lists: Access and Candidate. If a CHPID is either shared or reconfigurable, you can specify which partitions have access to that CHPID. Use the CHPID access list with or without the CHPID candidate list.

That is, CHPIDs are assigned to partitions by adding each partition to the CHPID's access or candidate list. A partition has initial access to a CHPID if the logical partition is on that CHPID's access list.

actual size. Actual Size scales the diagram so that objects are viewed at their approximate size when a diagram is printed at full size.

alternate processor. Alternate refers to a processor which is not the current logical view of its assigned machine.

B

button. Dialog option: click on button to carry out action. `OK`, `Cancel` and `Help` buttons appear in almost every dialog. Clicking `OK` will close the dialog. If you have more than one dialog open, click `OK` or `Cancel`, or press the `ESC` key repeatedly to close all dialogs and return to the diagram.

C

cabinet. Cabinets, also known as fiber management cabinets contain *panels* organized into *port groups* of *patchports*, which are pairs of fibre adapters or couplers. Cabinets are used to organize long, complex cables between processors and controllers, which may be as far away as different buildings. In the configuration diagram, cabinets are shown on the far right.

cable "in inventory". You can put unused cables *into inventory*; HCM will not "remember" the past usage of these cables. Cables in inventory are cables that are generally available for use to any object.

candidate list. A partition can gain access to a CHPID if the partition is on that CHPID's Candidate list. A

partition is allowed to configure a CHPID online if the logical partition is on that CHPID's Candidate list.

See also *access list*.

CFReport. When a machine is ordered, the output of the order process is a binary file that represents the physical description of the final machine. One of the components of that file is the type and physical location, including the Physical Channel Identifier (PCHID) value assigned to that location, of all the I/O features in the final machine. This file is called a CFReport.

channel adapter. A *channel adapter* groups two or more controller channel interfaces electronically.

channel interface. See [Controller] channel interface or [Crossbar] channel interface.

channel subsystem (CSS). A collection of subchannels that directs the flow of information between I/O devices and main storage. It uses one ore more channel paths as the communication link in managing the flow of information to or from I/O devices. Within the CSS is one subchannel set and logical partitions. One subchannel from the set is provided for and dedicated to each I/O device accessible to the CSS. Logical partitions use subchannels to communicate with I/O devices. The maximum number of CSSs supported by a processor depends on the processor type. If more than one CSS is supported by a processor, each CSS has a processor unique single hexadecimal digit CSS identifier (CSS ID).

CHPID. A logical processor contains a number of *CHPIDs*, or CHannel Path IDs, which are the logical equivalent of channels in the physical processor. See also:

- *dedicated CHPID*
- *reconfigurable CHPID*
- *shared CHPID*
- *spanned CHPID*

CHPID mapping. Usually, a processor can have up to 256 channels and 256 CHPIDs, so there is a 1:1 mapping. However, in processors of the "one processor per partition" style, there can be up to 512 channels or the relationship between CHPIDs and channels may not be one to one. The default is to map the CHPIDs to the first 256 channels, but you may specify a non-standard *CHPID mapping*.

CHPID Mapping Tool. The CHPID Mapping Tool aids the customer in developing a CHPID-to-PCHID relationship for XMP processors. It accepts an IOCP input file without PCHID values, allows the user to

assign the logical CHPID values in the input to the PCHIDs available with his ordered machine, and returns an updated IOCP input file that contains the PCHID values.

CHPID type. The logical *CHPID type* typically corresponds to a physical path interface type.

cluster. See *segment*.

CMT. See *CHPID Mapping Tool*.

config file. The *config file* (*.HCM file) contains HCM's interactive configuration diagram. You can modify and save different versions of your hardware configuration in a number of config files for historical and planning purposes. The config file is also called *work configuration file*. See also *work configuration file*, and *configuration files*.

configuration diagram. From the IODF on the host, HCM constructs an interactive *configuration diagram* to help you visualize and more easily maintain your system. You can add or modify objects in the diagram to more accurately represent your hardware configuration. You can also print out wall charts or reports to plan and implement future modifications to your system.

configuration files. The *configuration files* refer to work configuration files (*.HCM file) as well as to production configuration files (*.HCR file). They are equivalent to the work and production IODFs , but are located on the PWS. See also *work configuration file*, *config file*, and *production configuration file*.

context diagram. A *context diagram* is a configuration diagram showing the object of interest and all equipment connected to it. Think of a context diagram as the configuration as seen by this object.

cropped. A *cropped* diagram includes a subset of the entire configuration comprising the selected object(s) and any objects that they connect to.

controller. A *controller* translates high level requests from processors to low level requests to I/O devices, and vice versa. Each physical controller contains one or more logical *control units*, channel and device interfaces, and a power source.

Controller *channel interfaces* can connect to the CHPIDs on one or more processors; controller *device interfaces* can connect to the string interfaces on one or more strings of I/O devices.

Controllers can be divided into *segments*, or grouped into *subsystems.*

[controller] channel interface. A *controller* has a number of *channel interfaces* on the top which can be physically connected and logically defined to CHPIDs on processors, possibly via control unit interfaces on crossbar switches (if parallel / bus and tag), ports on

switches (if serial / ESCON or FICON), or converters (if connecting parallel <=> serial interfaces or vice versa). Note that the serial physical cables connecting the controller and processor(s) may travel through cabinets via patchports.

control unit. Each physical controller contains one or more *control units*, which translate high level requests to low level requests between processors and devices.

control unit interface. A crossbar switch has a number of control unit interfaces on its side which can connect to channel interfaces on controllers.

converter [P→S, S→P]. There are two types of *converter*: those that are used to connect serial ESCON cables to parallel bus-and-tag cables (9034 or S→P), and those that connect parallel to serial cables (9035, or P→S).

HCM automatically creates and connects converters when necessary. HCM maintains an inventory of converters currently in use or that were used for connections which have been subsequently disconnected.

coupling facility. The *coupling facility* is the hardware element that provides high speed caching, list processing, and locking functions in a sysplex. To enable data sharing between a CF partition and the central machines, special types of high speed CF channels are required. A CFR channel path attached to a CF partition can be connected to a CFS channel path attached to a partition in which an operating system is running. Besides CF connections between a CFR and CFS channel path, CF connections can also be established between two CFP channel paths.

[crossbar] channel interface. A crossbar switch has a number of *channel interfaces* on top which you can connect to CHPIDs either directly or via other crossbars.

crossbar configuration. You can document different settings of crossbar switch connectivity as *crossbar configurations*.

crossbar switch. A *crossbar switch* is a static switch that can connect controllers to processors with parallel (bus and tag) interfaces. The crossbar contains a number of *channel interfaces* on its top, which can connect to objects above it such as processors or other crossbars. The crossbar switch also contains a number of *control unit interfaces* on its side, which can connect to objects below it such as controllers or other crossbars.

crossbar switch connectivity. The *switch connectivity* comprises which crossbar channel interfaces are connected to which control unit interfaces; that is, which internal intersections in the crossbar are active.

CSS. See *channel subsystem*.

CTC connection (channel to channel connection).
You can create a CTC connection (channel-to-channel connection) between two CHPIDs on the same or different processors, either directly or through a switch. When connecting through a switch, both CHPIDs must be connected through the same or a chained switch. One CHPID must be CNC or FCV, the other CTC in case of an SCTC connection. For FCTC connections, two FC CHPIDs have to be connected.

HCM considers a CTC connection as a single entity comprising selected control units and I/O devices in the connected processors.

D

dedicated CHPID. A CHPID can be *dedicated* to one partition; only that partition can access I/O devices on this CHPID. All CHPID types can operate in DED (dedicated) mode.

device. See I/O Device.

device adapter. A *device adapter* groups two or more controller device interfaces electronically.

device interface. A *controller* has a number of *device interfaces* on the bottom which can connect to string interfaces on strings of I/O devices.

dialog. A *dialog* is an interactive pop-up window containing *options* which allow you to browse or modify information, take specific action relating to selected objects, or access other dialogs. HCM provides a series of dialogs to help you create, edit, delete, and connect objects, as well as manipulate the configuration diagram.

disconnect. A controller channel interface may be *disconnected* from a CHPID with either a *standard disconnection* or a *physical-only disconnection*. The standard disconnection involves logically disconnecting any control units from any CHPIDs that were only reachable via the controller channel interface, as well as physically disconnecting the controller from the processor.

drop down list box. Dialog option: click on the arrow icon, select an item from the list that drops down; your selection appears in the text box adjacent.

drop down box. Dialog option: click on the arrow icon, select an item from the list that drops down; your selection appears in the text box adjacent. Alternatively, type in the text box.

E

edit dialog. Type of dialog allowing you to edit data for a selected object.

ESCON switch. See switch.

extend-select. You can select more than one item in some list boxes. Hold down the SHIFT key while dragging the cursor over several sequential list box items (*extend selection*). See Multi-select.

F

fiber link. Fiber links are the physical fiber optic connections and transmission media between optical fiber transmitters and receivers. A fiber link can comprise one or more fiber cables and patchports in fiber management cabinets. Each connection in the fiber link is either permanent or mutable.

FICON switch. See switch

fiber management cabinet. See Cabinet.

fit to window. *Fit to Window* scales the diagram to completely fit in the display window.

full size. *Full size*, or 100% scale, is the largest scale for printing diagrams. At this size, each controller will be about one inch tall.

H

HCD. The Hardware Configuration Definition (*HCD*) is an interactive interface on the host which allows you to define the hardware configuration for both the channel subsystem and operating system.

head of string. The *head of string* is the first unit of devices in a string. It contains the string interfaces which connect to controller device interfaces.

I

independent controller. An *independent controller* is a subsystem which contains only that controller.

integrated device. Devices are said to be *integrated* into the control unit or controller if they are not attached to the controller by cables (for example, 3800 printers and TP lines defined in a 37x5).

interface type. The *interface type* is the physical path interface type usually corresponding to a particular CHPID type. Interface types include serial, parallel, coupling, IOC and OSA.

internal control units. See *switch's control unit*

internal devices. See *switch's devices*

internal structures. The controller and string structures used by HCM as containers for a switch's control units and devices are called the *internal structures*.

I/O device. An *I/O device* can be a printer, tape drive, hard disk drive, and so on. Devices are logically grouped inside *units*, which are in turn grouped into *strings*. The first unit, known as the *head of string*, contains *string interfaces* which connect to controller device interfaces and eventually to processor CHPIDs. Devices are represented as lines of text within the appropriate unit object in the configuration diagram.

IODF. The input/output definition file (*IODF*) file on the host contains processor and operating system I/O definitions, describing all logical objects and their connectivity in the hardware configuration.

J

jumper cable. Fiber *jumper cables* are usually used to make mutable connections between patchports.

L

LCSS. Logical channel subsystems. See also *channel subsystem*.

load [IODF load]. You can *load* IODF data from the host to the PWS, for HCM to use when creating the interactive configuration diagram for that hardware configuration. The selected IODF file is loaded into an empty PWS configuration.

logical definition. You can create logical definitions for objects which are physically connected and therefore, reachable. Logical definitions represent the connectivity between logical objects in the processor's logical view of the hardware configuration.

logical object. Logical objects belong to the processor's logical view of the hardware configuration. Generally, logical objects correspond to physical objects. For example, the logical processor has an associated physical processor (machine), and logical CHPIDs have corresponding physical channels. However, there are exceptions: logical partitions have no physical equivalent. Physical cabinets, patchports, or crossbar switches have no logical equivalents.

LPAR. See *partition*.

M

menu. A *menu* is a drop down list box containing *menu commands*. Click on a *menu title* in the menu bar at the top of the HCM screen to display the menu. Alternatively, hold down the ALT key and press the character key corresponding to the underlined letter in the menu title. For example, ALT+F for File menu.

Select a command by clicking on it; pressing the character key corresponding to the underlined letter in

the command, or; pressing up or down arrow keys until the desired command is highlighted, then pressing ENTER.

mode name. The name of the communication mode. This parameter is the same as the mode name on the host (logname). This name must match a value defined at the host.

multiple exposure device. A *multiple exposure device* is allocated by a single device number, but accessed by several device numbers, where each device number represents one *exposure*. The device number by which the device is allocated is the *base exposure*; all other device numbers are called *non-base exposures*.

multi-select. You can select more than one item in some list boxes. Press the CTRL key while clicking on list box items which may or may not be in sequence (*multi-select*). See Extend-select.

mutable connection. *Mutable connections* are usually made with fiber jumper cables between patchports in a cabinet or between cabinets and active objects such as CHPIDs, switches, converters and controllers with ESCON or FICON interfaces. Mutable connections are broken when the patchports they connect are not in use.

N

navigation dialog. Type of dialog allowing you to navigate to related dialogs.

O

online mode. The mode of operation when a connection to a host session (running HCD) is established for HCM.

option button. Dialog option: click on the button to select the option; all other options will be de-selected.

operation mode. The operation mode for CHPIDs defines how partitions may access I/O devices using this CHPID. The following operation modes are available:

- dedicated; see also *dedicated CHPID*
- reconfigurable; see also *reconfigurable CHPID*
- shared; see also *shared CHPID*
- spanned; see also *spanned CHPID*

option button. Dialog option: click on the button to select the option; all other options will be de-selected.

P

panel. Each cabinet can have one or more *panels*, arranged one above the other. Panels are organized into

port groups of *patchports*. They can be numbered starting from zero or one, from the top or the bottom of the cabinet.

panel space. A cabinet has a number of places to hold panels, called *panel spaces*.

partition. The logical processor may be divided into a number of logical *partitions* (LPARs). Each partition has access to the processor's resources for a limited amount of time. Partitions may be restricted from using certain CHPIDs and allowed to use others, by being assigned to each CHPID's Access or Candidate Lists. Restricting partitions to certain subsets of CHPIDs can provide system security; for example, company departments may each be assigned to a different partition.

patchport. A *patchport* comprises a pair of fibre adapters or couplers. Any number of patchports can participate in a fiber link. To determine the total number of patchports in a cabinet, you must add the number of patchports of each defined panel of the cabinet.

PCHID. See *physical channel identifier*.

peer-to-peer remote copy. PPRC (peer-to-peer remote copy) connections are direct connections between DASD controller subsystems that are used primarily to provide a hot standby capability. These connections can be point-to-point from one DASD controller to another, or they may pass through switches, just as connections from CHPIDs to control units can.

permanent connection. *Permanent connections* are usually made between cabinets with fiber trunk cables. Patchports that are permanently connected remain so even when they are not in use.

physical channel identifier (PCHID). The physical address of a channel path in the hardware. Logical CHPIDs have corresponding physical channels. Real I/O hardware is attached to a processor via physical channels. Channels have a physical channel identifier (PCHID) which determines the physical location of a channel in the processor. For XMP processors, the PCHIDs must be defined in the configuration. The PCHID is a three hexadecimal digit number and is assigned by the processor. One logical channel path (CHPID) provided by a channel subsystem may be associated with a physical channel (PCHID). There is no standard mapping between CHPIDs and PCHIDs. The CHPID Mapping Tool aids the customer in developing a CHPID-to-PCHID relationship. See also *CHPID Mapping Tool*.

physical description. A *physical description* is a template associated with a controller which describes its physical characteristics, for example, how to label controller interfaces. Most controllers can be associated with the generic physical description built into HCM. For those common IBM controllers unable to use the generic description, other physical descriptions are

provided. Sample templates or *PDFs* (physical description files) for other equipment are provided as examples to allow you to extend HCM's physical capabilities.

Physical descriptions are also used where required to handle unusual characteristics of physical strings containing I/O devices.

Note: The filename extension for physical description files was changed from ".pdf" to ".txt" for HCM 2.9 Service Level 3 and higher. Nevertheless, HCM 2.9 Service Level 3 and higher can still read physical description files having the extension ".pdf".

physical-logical inconsistency. A *physical / logical inconsistency* may occur when a physical route exists but a logical definition is missing, or when a logical definition exists without a physical pathway.

physical object. Physical objects exist; for example, the physical processor or machine has a serial number and a set of channels. Each physical object generally has a corresponding logical object; exceptions are physical cabinets, patchports and crossbar switches. You can add these physical objects to your configuration diagram after the IODF load, to more accurately portray your hardware configuration.

physical route. A physical *route* or *connection* provides a pathway between logical objects (such as control units and CHPIDs) that can be used by logical definitions.

port group. Panels are organized into *port groups* of *patchports*, arranged side by side.

PPRC. See peer-to-peer remote copy.

primary processor. *Primary* refers to a processor which is the current logical view of its assigned machine. At IODF Load / Resync time, the machine connections are built only for primary processors.

processor. The *physical processor*, or machine, has a serial number, a set of channels, and a *logical processor* associated with it. The logical processor has a number of channel path IDs, or *CHPIDs*, which are the logical equivalent of channels. The logical processor may be divided into a number of logical *partitions* (LPARs).

production configuration file. The *production configuration file* (*.HCR file) is the equivalent of the production IODF, but is located on the PWS. You are not allowed to modify the content of a production configuration file. The production configuration file and the production IODF form an associated pair. See also *configuration files*.

R

reconciliation of VOLSERs. The term *reconciliation* describes the process of synchronizing the VOLSER values in the PWS configuration file with those in the IODF.

reconfigurable CHPID. A *reconfigurable CHPID* is an unshared CHPID that you can reconfigure offline from one partition, then online to another. That is, the CHPID can be reconfigured between logical partitions after a power-on reset. Only one partition can access I/O devices on this CHPID at a time. All CHPID types can operate in REC (reconfigurable) mode.

report. You can specify certain parameters to compile and print HCM information as a *report*.

resync [IODF resync]. If an IODF file has been replaced or updated since its associated config file was saved, the IODF file and config file will be out of sync. HCM will resynchronize with the host IODF in these cases by reloading the IODF and making any appropriate physical changes.

S

SAID. See system adapter identifier.

section. Dialog option: groups similar dialog options together for easier viewing.

segment. Certain types of controller, such as the 3990, are divided into two separate sections, or *segments*. Note that in the 3990 controller, segments may be referred to as "storage clusters"; other controllers may refer to them as "busses". DASD controllers typically contain two segments; most other types of controller contain only one.

shared CHPID. A *shared CHPID* can be configured online to one or more partitions at the same time. One or more partitions can access I/O devices at the same time using this CHPID.

SMP processor. In this book, this term designates processors supporting a single channel subsystem. For SMP processors, the single channel subsystem is implicitly defined with the processor. This term is used in contrast to the term **XMP processor**, which designates processors supporting multiple logical channel subsystems.

spanned CHPID. With XMP processors, supporting multiple logical channel subsystems, some types of channel paths can be shared across partitions from multiple logical channel subsystems. It is dependent on the processor support, which channel types can be defined as spanned. Such a channel path is called a *spanned* channel path.

A spanned channel path will be created with the same CHPID number in all channel subsystems that are using it. For example, if you have a processor MCSSPRO1 with channel subsystems 0 through 3, and you create CHPID 1A (type IQD, SPAN) and let it access partitions from CSS 0, 2, and 3, you will get this CHPID 1A in CSSs 0, 2, and 3, but not in CSS 1.

SSID. See subsystem identifier.

SNA address. The system network architecture (*SNA*) address is used to identify the support element of a central processor complex (CPC) configured in a S/390 microprocessor cluster. The SNA address comprises the name of the network to which the CPC is connected, and the system name identifying the CPC within the network. The SNA address is part of the processor definition for a CPC, associating the processor defined in the IODF with the CPC configured in the S/390 microprocessor cluster.

standalone mode. This mode of operation of HCM allows you to open a configuration without a connection to the host.

string. A *string* refers to a collection of one or more I/O devices. The term usually refers to a physical string of units, but may mean a collection of I/O devices which are integrated into a control unit.

string interface. A string has a number of *string interfaces* on the top which connect to controller device interfaces, and eventually to processor CHPIDs.

subsystem. Some controller types can be grouped together in a *subsystem*. For example, 3990 dual frame controllers or 3803s linked via communications cables

subsystem identifier. A subsystem identifier (SSID) represents the four-digit hexadecimal number that is programmed into the hardware when it is installed, which uniquely identifies a logical control unit to a host system. For HCM purposes, the SSID is only required for integration with other PPRC tools (like RCMF). Typically, each DASD logical subsystem involved in a PPRC connection is labeled with a unique SSID, but it is usually not applied to non-DASD logical subsystems.

switch. A device that provides connectivity capability and control for attaching any two ESCON or FICON links together.

switch configuration. You can create a set of connection rules called a *switch configuration*, which can be used to determine which switch ports are connected internally (activated) at any one time. That is, you can dynamically restrict connections between switch ports. Each switch configuration deems that only certain connections are "on"; that is, only certain intersections between internal switch ports are active. Any switch port may connect to any other switch port.

The control unit inside the switch can receive messages from the processor, dynamically setting the configuration for the switch ports at that time. HCM allows you to create, view, edit or delete switch configurations.

switch connectivity. The *switch connectivity* comprises the internal connections between switch ports. Switch connectivity is dynamically set by switch configurations.

switch controller. The *switch controller* contained in an ESCON or FICON switch receives and interprets processor messages to set the switch configuration dynamically. This controller is not drawn in the configuration diagram.

The switch controller contains the *switch's control unit*, which is always connected to its switch port with the address FE. The switch's control unit is connected to a *device*, also within the switch. You can edit the control unit and device as you would normally.

switch's control unit. The *switch's control unit* resides in the controller within a switch. It can receive messages from the processor, dynamically setting the configuration for the switch's ports at that time. This control unit is always connected to the switch port with the address FE, and to a device, also in the switch.

switch's devices. The devices which are defined to the switch's control unit to control a switch are called switch's devices, they are internal to the switch. See also *switch controller*.

switch port. A switch port is an interface on the switch which can connect to a processor CHPID either directly or via another switch, or to a controller channel interface.

symbolic destination name. The symbolic destination name (SDN) relates the host name known to you to a partner LU (Windows 9x/NT), mode name (VTAM), and TP name (APPC/MVS).

system adapter identifier. A system adapter identifier (SAID) represents the two or four-digit hexadecimal number that is programmed into the hardware when it is installed, which uniquely identifies a physical channel interface on the controller.

T

text box. Dialog option: click in a text box and type in data. Use TAB or SHIFT-TAB to move between text boxes in a dialog.

trunk cable. *Trunk cables* are generally used to make permanent connections between cabinets which remain even when not in use.

U

unit. A *unit* is a collection of I/O devices in a physical box. There can be several units in a string; the first unit, known as the *head of string*, contains *string interfaces* which connect to controller device interfaces.

unused cable. *Unused cables* are physical cables that have been recently disconnected, but not yet placed in inventory. Think of them as lying on the floor, ready to be reused. Unused cables "remember" where they were last used.

V

validated work IODF. A validated work IODF satisfies all validation rules for building production IODFs. It may lack physical channel identifiers (PCHIDs) for XMP processors. In cooperation with HCD and the CHPID Mapping Tool a validated work IODF is required to accept new or updated PCHIDs. From such a validated work IODF, an IOCP input deck suitable for the 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.

W

work configuration file. The *work configuration file* (*.HCM) contains HCM's interactive configuration diagram. You can modify and save different versions of your hardware configuration in a number of work configuration files for historical and planning purposes. The work configuration file and the work IODF form an associated pair. See also *config file* and *configuration files*.

X

XMP processor. In the S/390 context, this term designates processors that support multiple logical channel subsystems (LCSS). It is used in contrast to the term SMP processor, which designates processors of previous generations that support only one channel subsystem. In general, the different CSSs including their channel paths and logical partitions provided by an XMP processor operate independently from each other. Channel paths can be spanned over multiple logical channel subsystems on the same processor depending on the channel path type. See also *SMP processor* and *channel subsystem*.

Index

Numerics

A

B

C

Readers' Comments — We'd Like to Hear from You

z/OS and z/VM
Hardware Configuration Manager
User's Guide

Publication No. SC33-7989-05

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?   ☐ Yes   ☐ No

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.

Name _____   Address _____

Company or Organization _____

Phone No. _____

IBM ®

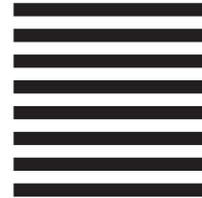Fold and Tape   **Please do not staple**   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape   **Please do not staple**   Fold and Tape

SC33-7989-05

IBM ®

Program Number: 5694-A01, 5655-G52, 5741-A05

Printed in USA